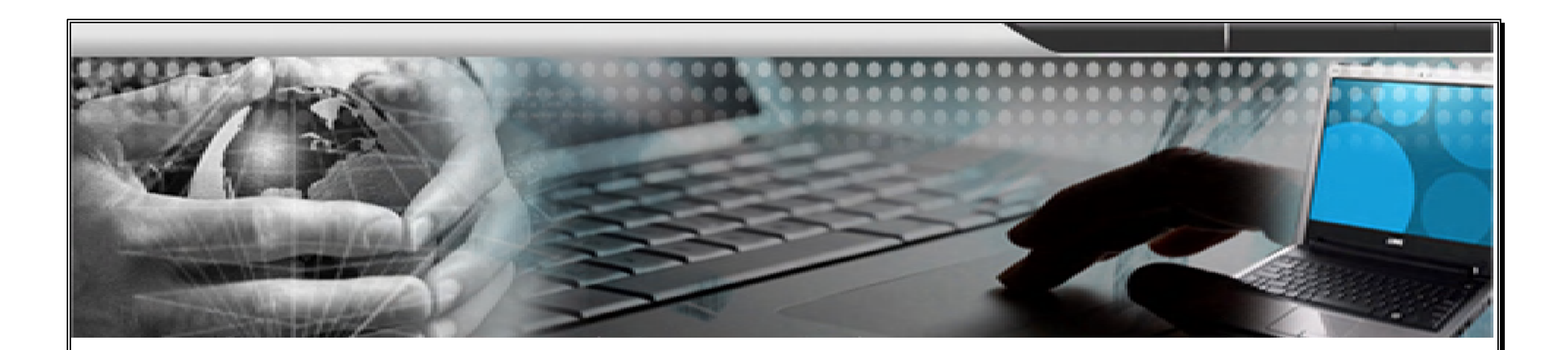


بسم الله الرحمن الرحيم

Look Like Microsoft Visual C++

Explanation Microsoft Visual C++ Tools

إعداد و برمجة

عبدالكريم محمد أبونمر

""سبحانك لا علم لنا إلا ما علمتنا إنك أنت العليم الحكيم "" صدق الله العظيم

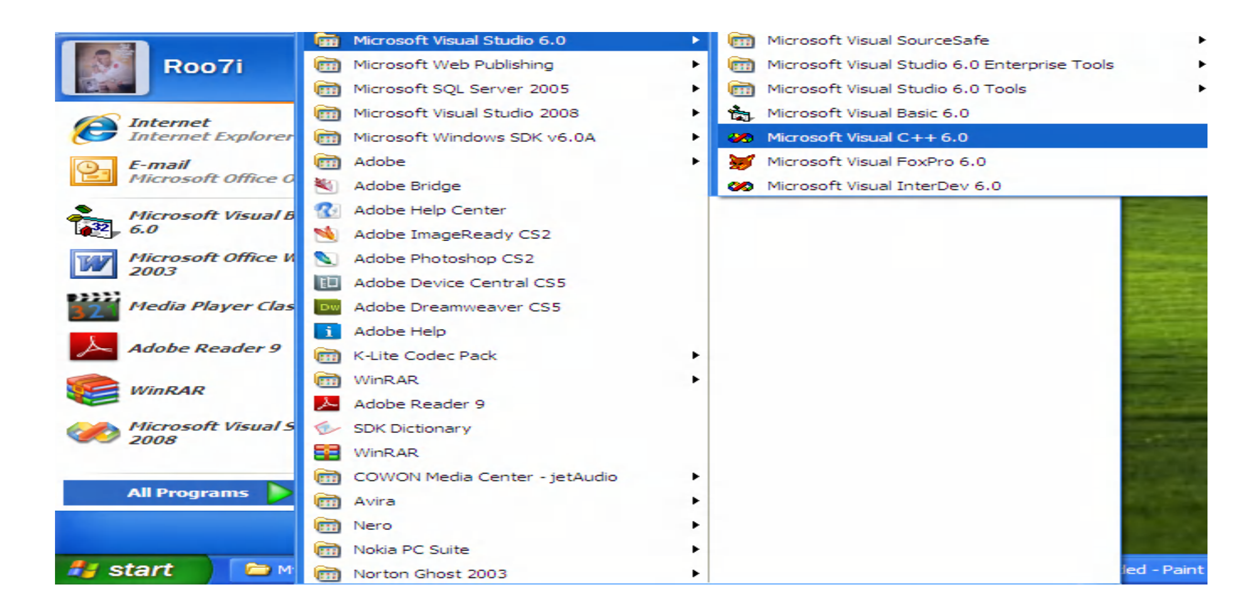
أقدم لكم كتابي المتواضع متمنياً فيه رضا المولى عز و جل .

** موضوعات الكتاب

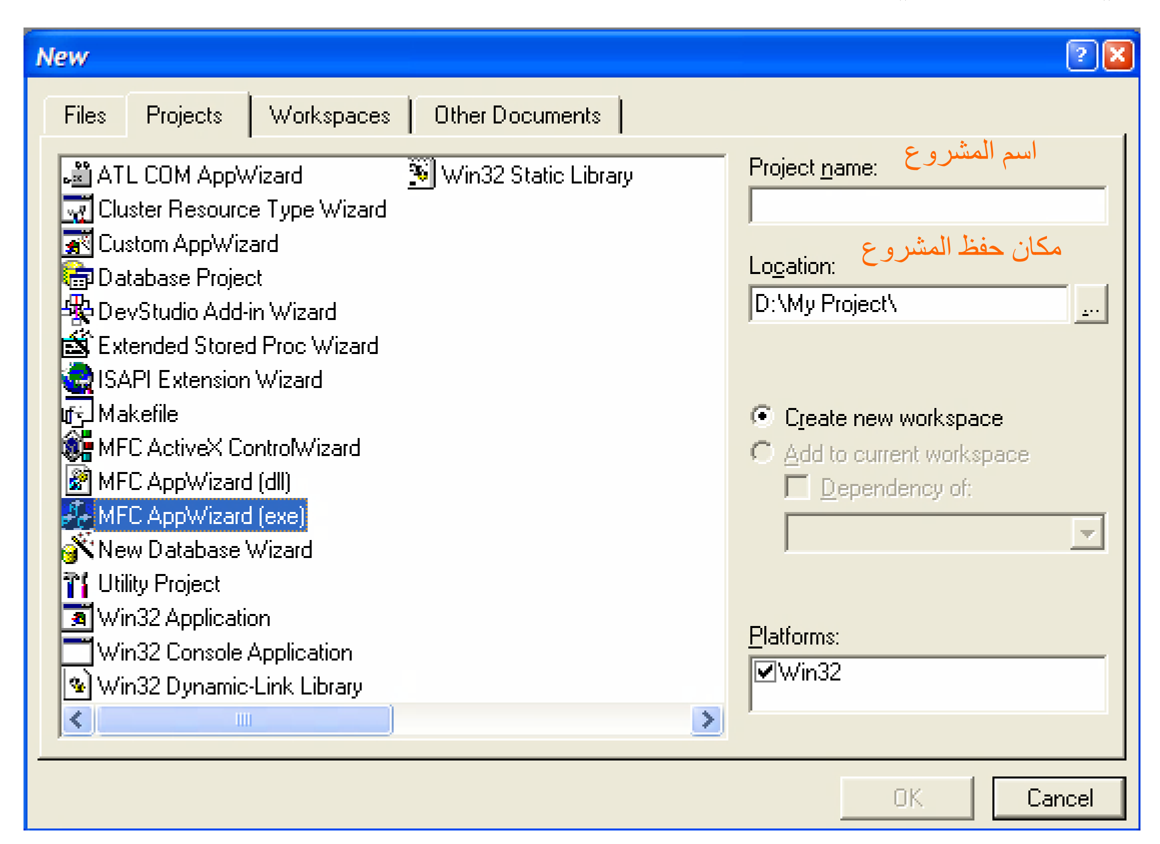
* Edit Box
* Radio Box & Check Box
* Group Box
* List Box
* Combo Box
* Design New Dialog
* Type of Message
* Single Document Interface
* Working with Mouse Events
* Working with Keyboard Events
* Cclint DC Class
* Graphics (Pen / Font / Paint)
* Timer
* Advanced Control
* Active X Control
* Toolbar & Status Bar
* Files
* Printing
* Help Files

كيفية إنشاء برنامج بلغة ++VC قم بإتباع الصور التالية :

1. فتح برنامج ++Microsoft Visual C

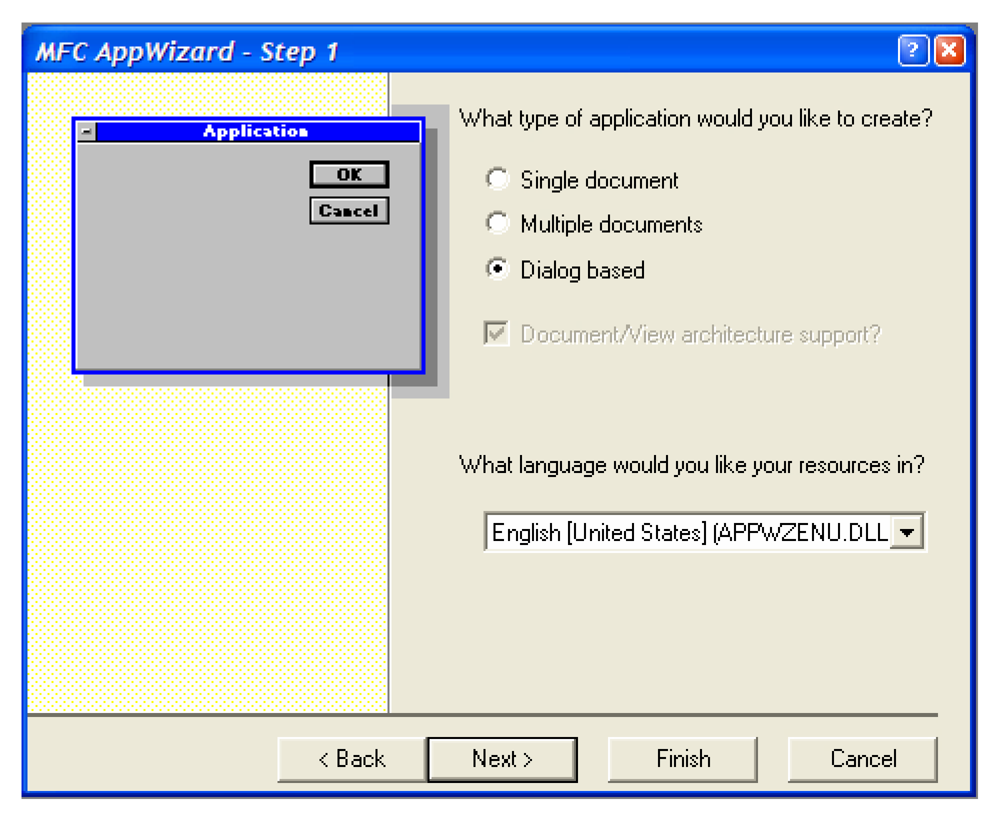


2. من لائحة ملف File نختار أمر جديد New فيظهر صندوق حوار New كما في الشكل التالي

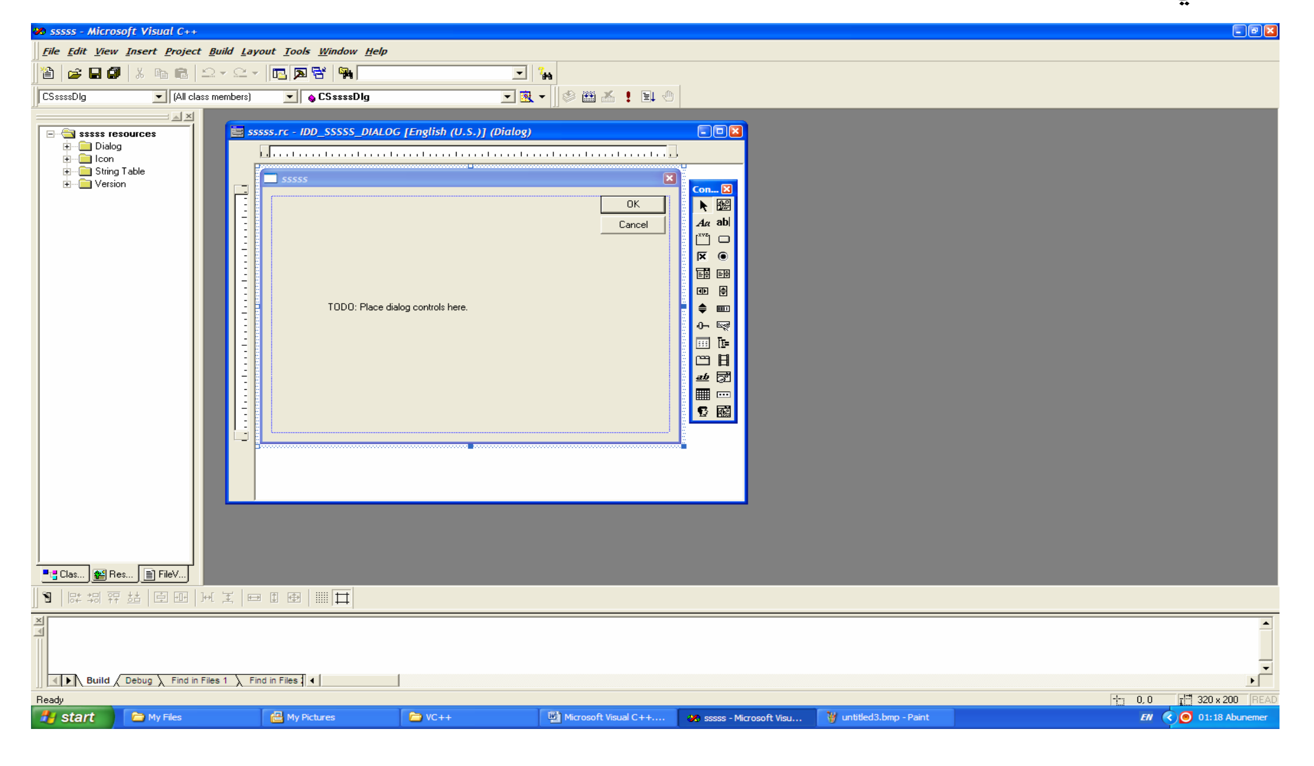


نختار من تبويب Projects الخيار ((MFC AppWizard(exe))

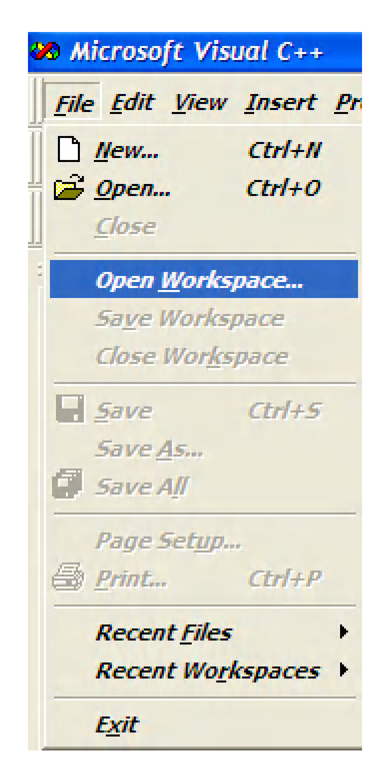
ثم نكتب اسم المشروع و مكان حفظه ثم نضغط على زر OK
3. يظهر صندوق حوار نختار منه خيار Dialog based كما في الشكل التالي

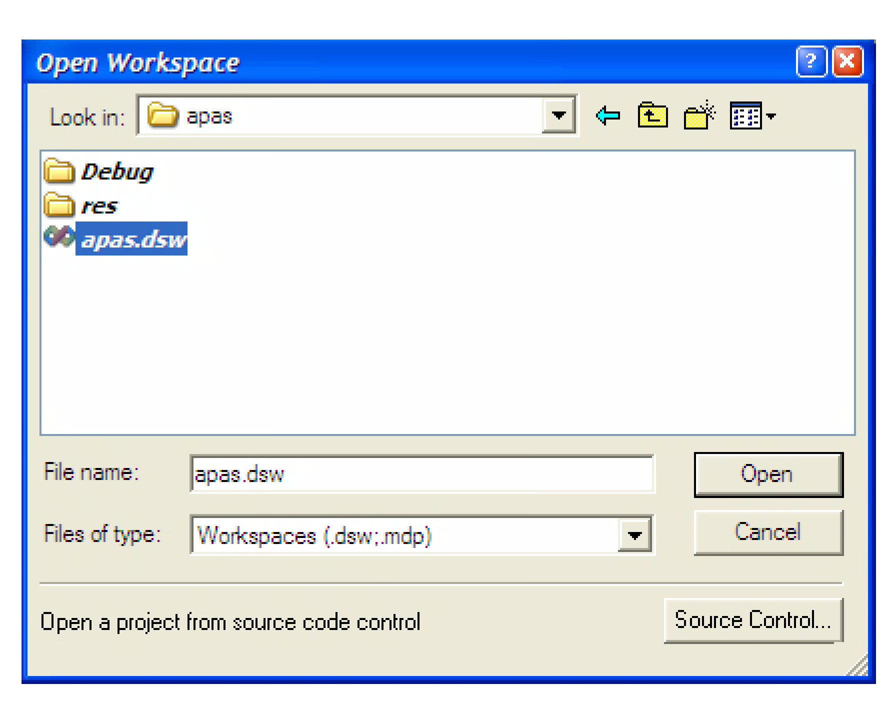


ثم نضغط على زر أمر Finish . فتظهر شاشة البرنامج كما في الشكل التالي



** فتح ملف منشئ مسبقاً
من لائحة File نختار Open Workspace و اختيار الملف الذي تريد فتحه بإمتداد (.dsw) كما في الشكل التالي :





ثم قم بالضغط على زر أمر Open .

** Microsoft Visual C++ Tools

* Edit Box Tool

- تستخدم لأدخال و إخراج البيانات و تسمى (IO Tools)

* من خصائص هذه الاداة :-

1. محاذاة النص Align و قيمه (Right , Left , Center)
2. أستخدام نص الكتابة لادخال رقم السري(Password)
3. تعدد الأسطر أو تعدد الأعمدة أفقي (Horizontal) عمودي (Vertical) .
4. الكتابة بوضع (Capital Letter / Small Letter) .
5. إظهار الكتابة بشكل Invisible .

** ملاحظة مهمة : للتعامل مع أي أداة يجب تعريف متغير لها

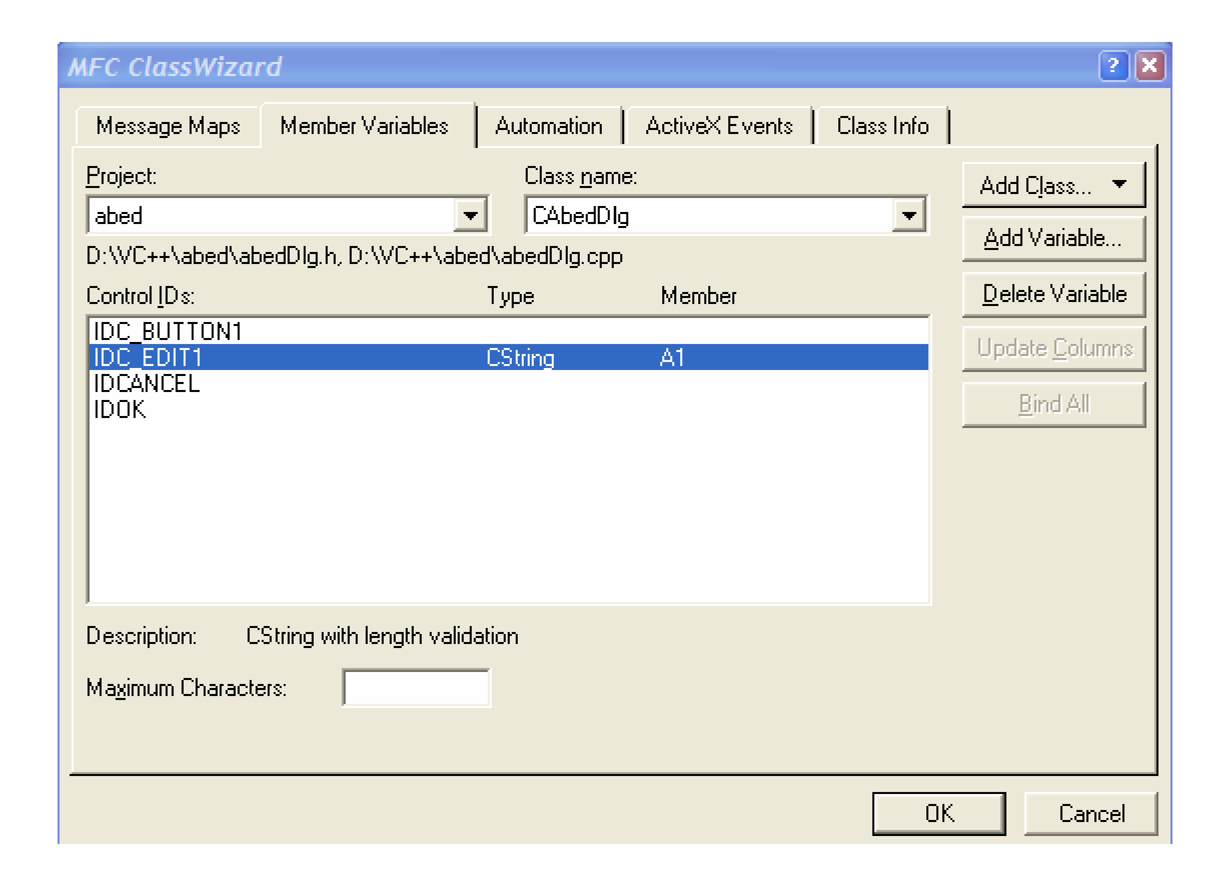
* Caption & ID : تستخدمان لتسمية Edit Box
- Caption :- الاسم الذي يظهر للمستخدم و يمكن تغييره .
- ID :- يستخدمها المبرمج في لغة البرمجة و لا يمكن تغييره .

- (UpdateData(True)) تستخدم للكتابة في مربع النص .
- (UpdateData(False)) تستخدم لأظهار الكتابة على مربع النص .

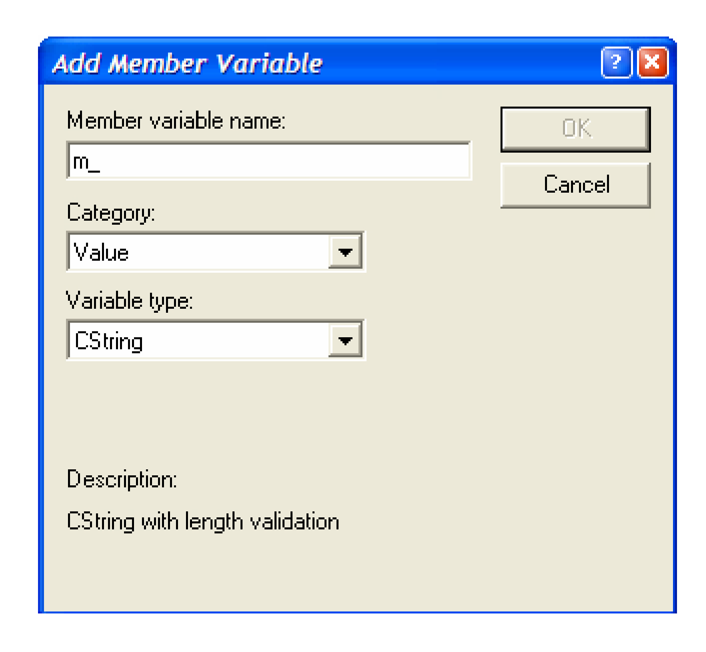
* مكان تعريف المتغير

- لتعريف متغير للاداة نقوم بالخطوات التالية :

1. الذهاب الى قائمة View و منها نختار Class Wizard فيظهر صندوق الحوار كما في الشكل التالي



نختار تبويب Member Variables و نقوم بتحديد الاداة و الضغط على زر أمر Add Variables فيظهر الصندوق الحوار التالي :



في صندوق النص الاول يتم وضع اسم المتغير

* Category:

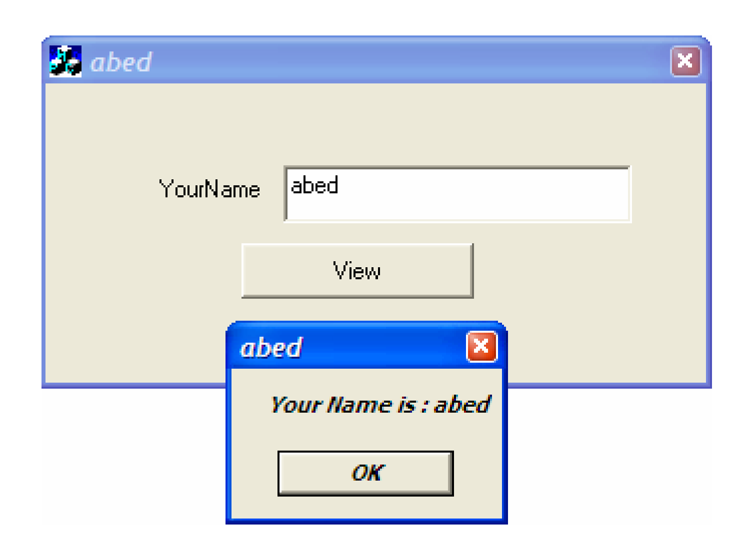
- Value: لا يستطيع أن يتعامل مع دوال الاداة
- Control: يستطيع أن يتعامل مع دوال الاداة

* Variables Type:

نوع بيانات المتغير

ومن ثم الضغط على زر امر OK .

*** مثال : قم بعمل برنامج يقوم المستخدم بكتابة أسمه في مربع النص و عند الضغط على زر أمر يقوم البرنامج اسم المستخدم في رسالة .**

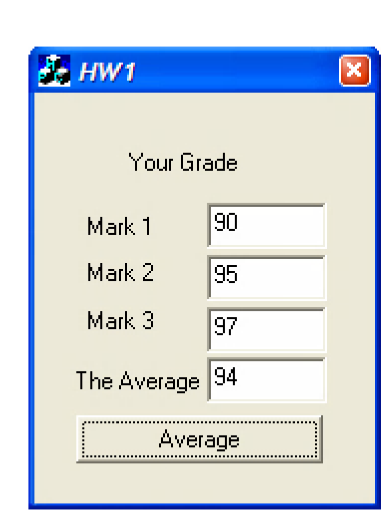



```
void CAbedDlg::OnButton1()
{
    UpdateData(true);
    MessageBox ("Your Name is :  " + A1);

}
```

* مثال : قم بعمل برنامج لحساب معدل الطالب لثلاث علامات

```
void CHW1Dlg::OnButton1()
{
    UpdateData(true);
    m4=(m1+m2+m3)/3;
    UpdateData(false);
}
```



** Radio Box & Check Box Tools

* Check Box:

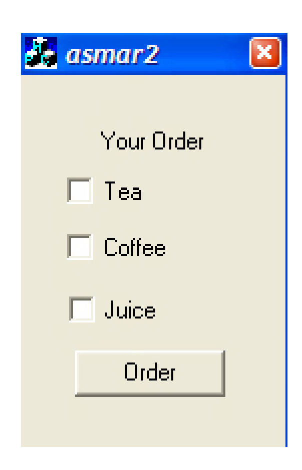
أختيار أكثرمن خيار في نفس الوقت

يجب تعريف متغير للأداة من نوع Bool

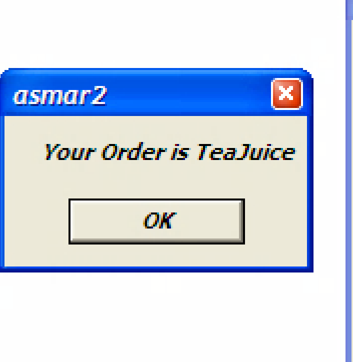
* وظائفه

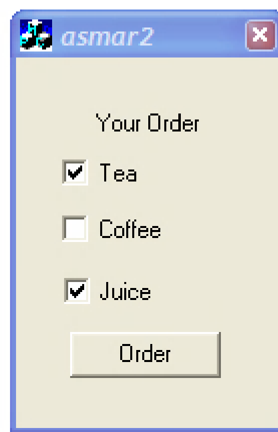
1. لفحص الاداة أنها تم اختيارها أم لا (Checked/ Un Checked)
2. لإظهار الاداة و اخفائها (Show / Hide) .
3. لتفعيل أو تعطيل الاداة (Enabled / Disabled) .

مثال : قم بعمل البرنامج التالي



وعند أختيار أي طلب يتم عرضها في رسالة بعد الضغط على زر أمر Order



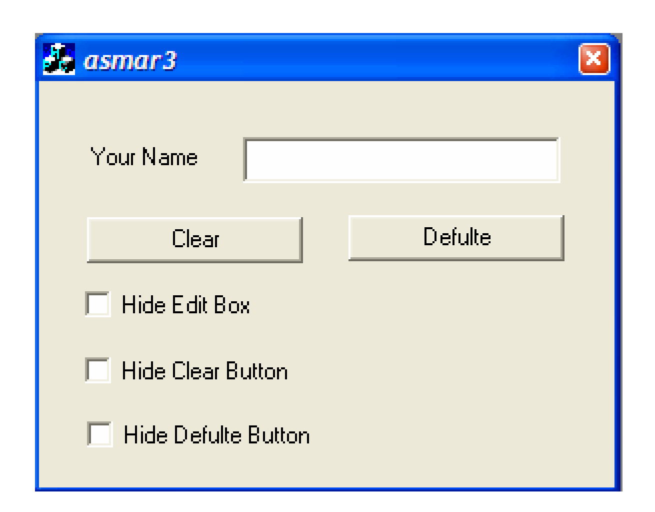



```
void CAsmar2Dlg::OnButton1()
{
     CString Str;
     UpdateData(true);
     if (m1==true)
          Str+="Tea";
     if (m2==true)
          Str+="Coffee";
     if (m3==true)
          Str+="Juice";
     UpdateData(false);
     MessageBox("Your Order is " + Str);
}
```

** ملاحظة : يجب استخدام UpdataData( ) أيضاً مع CheckBox

- Str = "Tea";
يحذف أي شيء موجود في المتغير و تخزن القيمة الجديدة
- Str+="Tea";
يضيف القيمة الجديدة للقيمة القديمة الموجودة في المتغير

* مثال : قم بتصميم الشكل التالي :



عمل البرنامج

1. عند الضغط على زر Clear يتم مسح ما في صندوق النص
2. عند الضغط على زر Default يتم إظهار كلمة Welcome في صندوق النص .
3. عند تفعيل الخيار الاول يتم أخفاء صندوق النص و عند ازالة التفعيل يتم إظهار صندوق النص
4. عند تفعيل الخيار الثاني يتم أخفاء زر أمر Clear و عند ازالة التفعيل يتم إظهار زر الامر
5. عند تفعيل الخيار الثالث يتم أخفاء زر أمر Default و عند ازالة التفعيل يتم إظهار زر الامر

```
void CAsmar3Dlg::OnButton1()
{
	e1= "";
	UpdateData(false);
}

void CAsmar3Dlg::OnButton2()
{
	e1="Welcome";
	UpdateData(false);

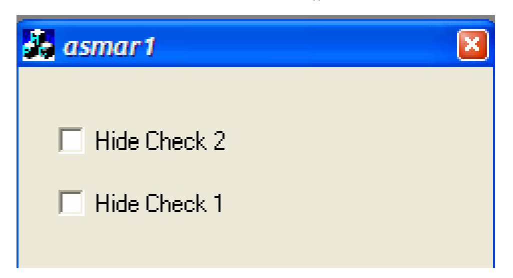
}

void CAsmar3Dlg::OnCheck1()
{
	UpdateData(true);
	if (m1==true)
		GetDlgItem(IDC_EDIT1)->ShowWindow(SW_HIDE);
	else
		GetDlgItem(IDC_EDIT1)->ShowWindow(SW_SHOW);
	UpdateData(false);
}

void CAsmar3Dlg::OnCheck2()
{
	UpdateData(true);
	if (m2==true)
		GetDlgItem(IDC_BUTTON1)->ShowWindow(SW_HIDE);
	else
		GetDlgItem(IDC_BUTTON1)->ShowWindow(SW_SHOW);
	UpdateData(false);
}

void CAsmar3Dlg::OnCheck3()
{
	UpdateData(true);
	if (m3==true)
		GetDlgItem(IDC_BUTTON2)->ShowWindow(SW_HIDE);
	else
		GetDlgItem(IDC_BUTTON2)->ShowWindow(SW_SHOW);
	UpdateData(false);
}
```

مثال : قم بعمل الشكل التالي :



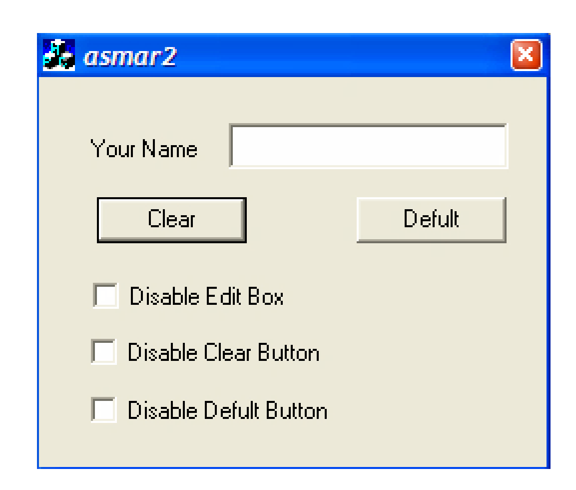
عمل البرنامج

1. عند تفعيل الخيار الاول يتم إخفاء الخيار الثاني و عند إزالة التفعيل يتم إظهار الخيار الثاني .
2. عند تفعيل الخيار الثاني يتم إخفاء الخيار الاول و عند إزالة التفعيل يتم إظهار الخيار الاول .

```
void CAsmar1Dlg::OnCheck1()
{
	UpdateData(true);
	if (m1==true)
		GetDlgItem(IDC_CHECK2)->ShowWindow(SW_HIDE);
	else
		GetDlgItem(IDC_CHECK2)->ShowWindow(SW_SHOW);
	UpdateData(false);
}

void CAsmar1Dlg::OnCheck2()
{
	UpdateData(true);
	if (m2==true)
		GetDlgItem(IDC_CHECK1)->ShowWindow(SW_HIDE);
	else
		GetDlgItem(IDC_CHECK1)->ShowWindow(SW_SHOW);
	UpdateData(false);
}
```

مثال : قم بعمل الشكل التالي :



عمل البرنامج

5. عند الضغط على زر Clear يتم مسح ما في صندوق النص
6. عند الضغط على زر Default يتم إظهار كلمة Welcome To VC++ في صندوق النص .
7. عند تفعيل الخيار الاول يتم تعطيل صندوق النص و عند ازالة التفعيل يتم تفعيل صندوق النص
8. عند تفعيل الخيار الثاني يتم تعطيل زر أمر Clear و عند ازالة التفعيل يتم تفعيل زر الامر
9. عند تفعيل الخيار الثالث يتم تعطيل زر أمر Default و عند ازالة التفعيل يتم تفعيل زر الامر

```
void CAsmar2Dlg::OnButton1()
{
	E1="";
	UpdateData(false);
}

void CAsmar2Dlg::OnButton2()
{
	E1="Welcome To VC++";
	UpdateData(false);
}

void CAsmar2Dlg::OnCheck1()
{
	UpdateData(true);
	if (m1==true)
		GetDlgItem(IDC_EDIT1)->EnableWindow(false);
	else
		GetDlgItem(IDC_EDIT1)->EnableWindow(true);
	UpdateData(false);
}

void CAsmar2Dlg::OnCheck2()
{
	UpdateData(true);
	if (m2==true)
		GetDlgItem(IDC_BUTTON1)->EnableWindow(false);
	else
		GetDlgItem(IDC_BUTTON1)->EnableWindow(true);
	UpdateData(false);
}

void CAsmar2Dlg::OnCheck3()
{
	UpdateData(true);
	if (m3==true)
		GetDlgItem(IDC_BUTTON2)->EnableWindow(false);
	else
		GetDlgItem(IDC_BUTTON2)->EnableWindow(true);
	UpdateData(false);

  }
```

* Radio Box

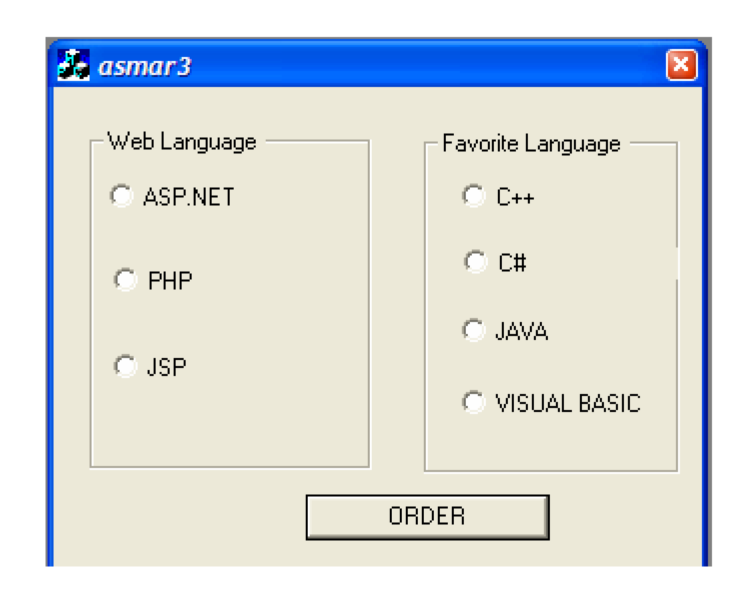
لا تستطيع أختيار أكثر من خيار في وقت واحد
نستخدم مع هذه الاداة (Group Box) و تستخدم هذه الاداة لعزل جميع الادوات التي بداخلها عن بقية الادوات و تأخذ الادوات التي بداخله نفس الخصائص .

يجب تعريف متغير للأداة من نوع Int .

* طريقة إضافة الادوات لـ Group Box :
1. تنزيل الـ Group Box و تسميته .
2. تنزيل الادوات بداخل الـ Group Box
3. الذهاب الى خصائص أول أداة و تفعيل خاصية الـ Group .

** ملاحظة :
- إذا اضفت Radio Box بدون الـ Group Box رح نتعامل مع كل أداة على حدة .
- إذا أضفت Radio Box داخل الـ Group Box رح نتعامل مع الادوات على أنها أداة واحدة فقط .

* مثال : قم بعمل الشكل التالي :



عمل البرنامج
عند أختيار اللغات المفضلة و الضغط على زر الامر يتم عرضها في رسالة .

```
void CAsmar3Dlg::OnButton1()
{
	CString Str;
	UpdateData(true);
	if (R1==0)
		Str+="ASP.NET\n";
	if (R1==1)
		Str+="PHP\n";
	if (R1==2)
		Str+="JSP\n";
	if (R2==0)
		Str+="C++\n";
	if (R2==1)
		Str+="C#\n";
	if (R2==2)
		Str+="JAVA\n";
	if (R2==3)
		Str+="VISUAL BAISC\n";
	MessageBox("Your Order is " + Str);
	UpdateData(false);
}
```

* **List Box Tools**

أداة تستخدم لعرض مجموعة من الاختيارات
نستطيع إضافة عنصر لها ، حذف عنصر ، إختيار عنصر معين .
و يمكن إعطائها عناصر إبتدائية .
يجب تعريف متغير للأداة و عند تعريف المتغير يجب أن نجعل المتغير يستطيع التعامل مع دوال الأداة فالذا يجب أن نختار من
Category = Control

* خصائصها

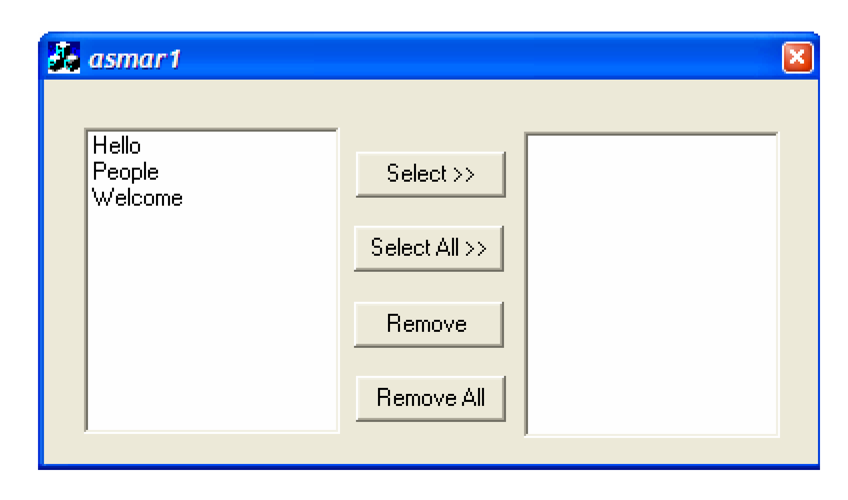
1. Visible
2. Disable
3. Selection و يأخذ القيم التالية :

أ. Single : يستطيع المستخدم أختيار عنصر واحد
ب. Multiple : يستطيع المستخدم أختيار أكثر من عنصر .
ج. Extend : يستطيع المستخدم أختيار أكثر من عنصر مع الضغط على مفتاح Shift + Ctrl .
د. Noun : لا يستطيع المستخدم أن يختار اي عنصر .
هـ. Sort : ترتيب العناصر إذا كانت قيمتها True يرتب العناصر أبجدياً و False لا يرتب العناصر و العنصر المضاف يضاف الى أخر القائمة .

* دوال الــ List Box :

1. AddString(" ") : إضافة النص الى القائمة .
2. InsertString(index," ") : إضافة النص الى القائمة بموقع معين .
3. ResetContents() : حذف جميع العناصر .
4. GetText() : ترجع العنصر المختار
5. GetCursel() : ترجع موقع العنصر الذي أختاره المستخدم من القائمة .
6. ShowWindow() : true : إخفاء مربع السرد / Falseإظهار مربع السرد
7. DeleteString(index) : حذف العنصر في موقع محدد
8. FindString(index," ") : للبحث عن عنصر مع تحديد موقع بداية البحث و إذا لم يجد العنصر يرجع -1
9. GetCount() : ترجع عدد العناصر في القائمة .
10. GetTextLen(index) : ترجع طول العنصر في الموقع المحدد

* مثال : قم بعمل الشكل التالي :




```
BOOL CAsmar1Dlg::OnInitDialog()
{
	CDialog::OnInitDialog();
	L1.AddString("Hello");
	L1.AddString("Welcome");
	L1.AddString("People");

		// Add "About..." menu item to system menu.


void CAsmar1Dlg::OnButton1()
{
	int m;
	CString a;
	m= L1.GetCurSel();
	L1.GetText(m,a);
	L2.AddString(a);
}

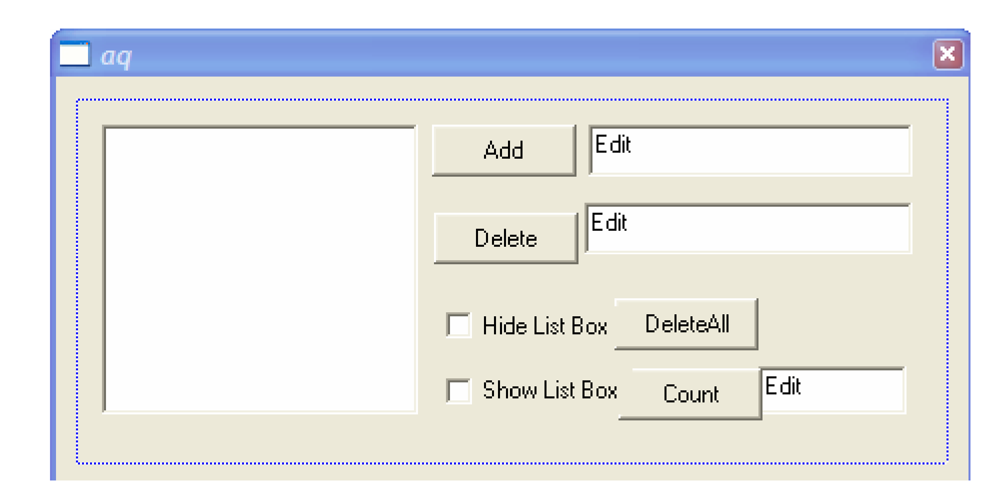
void CAsmar1Dlg::OnButton2()
{
	CString m;
	for(int i=0;i<L1.GetCount();i++)
	{
		L1.GetText(i,m);
		L2.AddString(m);
	}
}
```

```
void CAsmar1Dlg::OnButton3()
{
	L1.DeleteString(L1.GetCurSel());

}

void CAsmar1Dlg::OnButton4()
{
	L1.ResetContent();
}
```

* مثال قم بعمل الشكل التالي :



عمل البرنامج

1. عند الضغط على زر Add يتم إضافة النص الموجود في صندوق النص الاول .
2. عند الضغط على زر Delete يتم حذف العنصر المحدد في صندوق النص الثاني
3. عند تفعيل الخيار الاول يتم إخفاء List Box
4. عند تفعيل الخيار الثاني يتم إظهار List Box
5. عند الضغط على زر DeleteAll يتم حذف جميع العناصر
6. عند الضغط على زر Count يتم إظهار عدد العناصر في صندوق النص الثالث

```
void CAqDlg::OnButton1()
{
	UpdateData(true);
	L1.AddString(A1);
	UpdateData(false);
}

void CAqDlg::OnButton2()
{
	UpdateData(true);
	L1.DeleteString(A2);
	UpdateData(false);
}

void CAqDlg::OnCheck1()
{
	UpdateDate(true);
	if (a1==true)
		GetDlgItem(IDC_LIST1)->ShowWindow(SW_HIDE);
	UpdateData(false);
}

void CAqDlg::OnCheck2()
{
	UpdateDate(true);
	if (a2==true)
		GetDlgItem(IDC_LIST1)->ShowWindow(SW_SHOW);
	UpdateData(false);
}

void CAqDlg::OnButton3()
{
	L1.ResetContent();
}

void CAqDlg::OnButton4()
{
	UpdateData(true);
	A3= L1.GetCount();
	UpdateData(false);
}
```

*Combo Box Tools

هو عبارة عن List Box الجزء الاسفل الذي يظهر عند النقر على السهم . و هو عبارة عن Edit Box الجزء العلوي (أستطيع الكتابة عليه) .

يجب تعريف متغير للأداة و عند تعريف المتغير يجب أن نجعل المتغير يستطيع التعامل مع دوال الأداة فالذا يجب أن نختار من

Category = Control

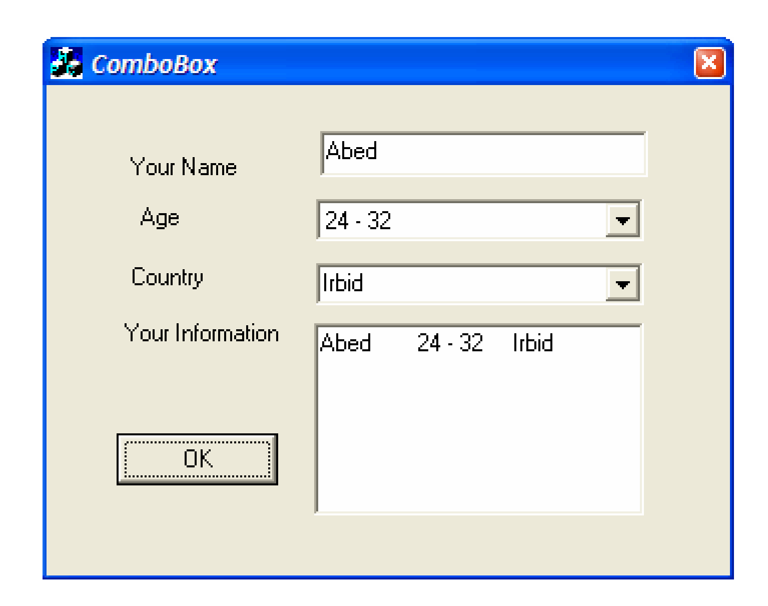
**** خصائص الـ Combo Box :**

1. **Type ——— Drop Down: أستطيع الكتابة في الجزء العلوي / Drop List: لا أستطيع الكتابة في الجزء العلوي.**
2. **Sort (الترتيب) : T : ترتيب العناصر أبجدياً / F : لا يرتب العناصر أبجدياً**
3. **Data : إعطاء قيم أبتدائية للاداة كما يلي :**
   Jordan , Japan , Syria
   للفصل بين القيم نضع فاصلة بينهم أو بعد كل قيمة نضغط على مفتاح Ctrl+Enter

**** دوال الـ Combo Box :**

نفس عمل دوال أداة الـ List Box لا فرق بينهما .

* مثال : قم بعمل البرنامج التالي :



عمل البرنامج
يتم كتابة اسم المستخدم في الخانة المخصصة لها و اختيار عمره و المنطقة السكنية له و عند الضغط على زر OK يتم عرض بياناته في Edit Box2 .

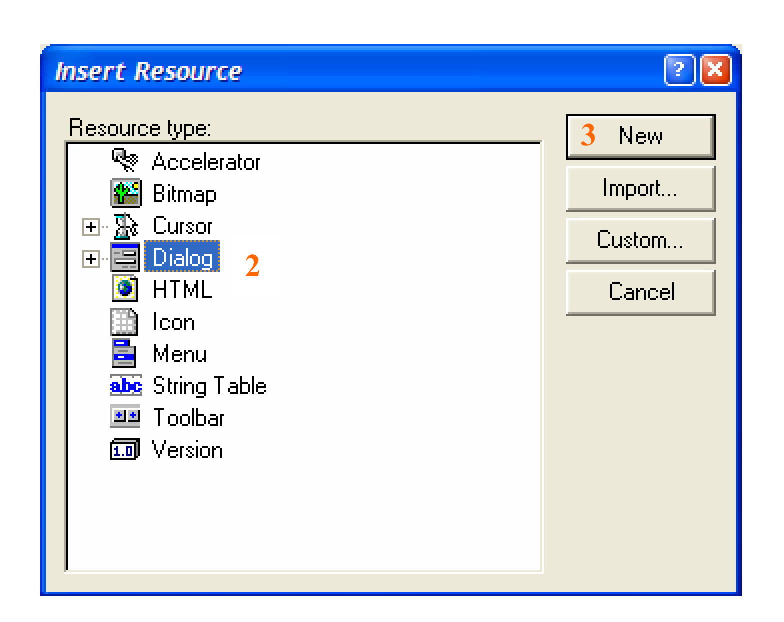
```
{
    UpdateData(true);
    A1+=A2;
    A1+="\t";
    char c[50];
    C2.GetWindowText(c,20);
    A1+=c;
    A1+="\t";
    C3.GetWindowText(c,20);
    A1+=c;
    A1+="\t";
    UpdateData(false);
}
```

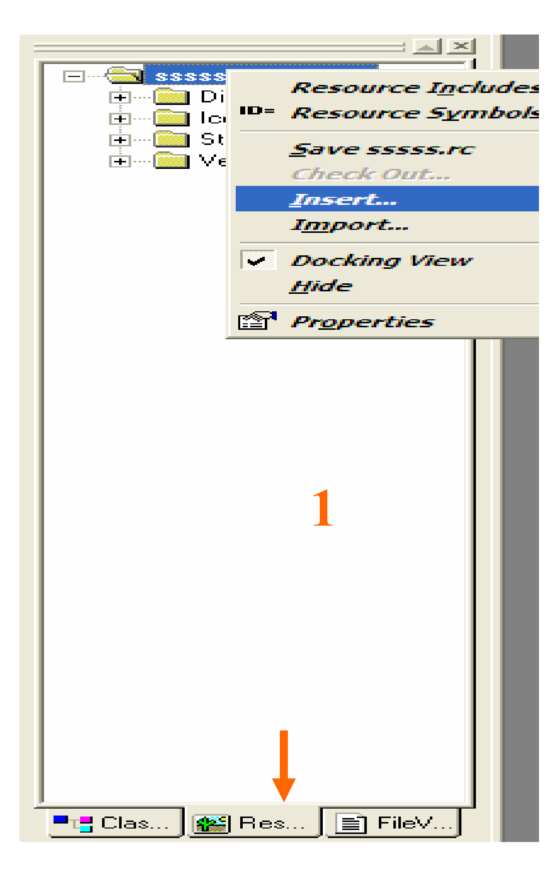
* Design New Dialog

تصميم فورم جديد

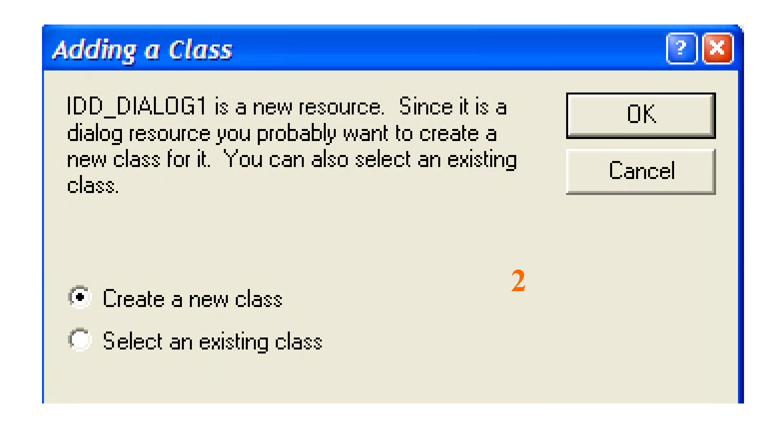
* خطوات تصميم فورم جديد :

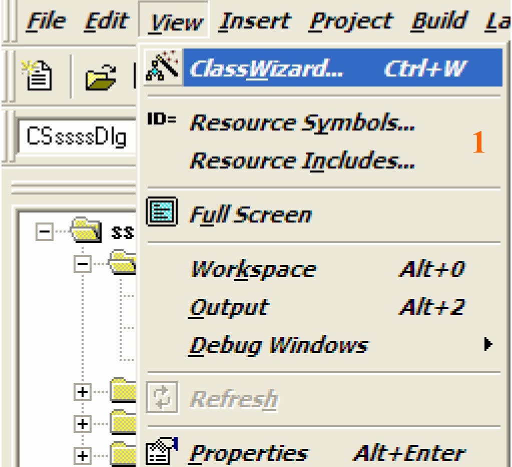
1. نذهب الى Resource و منها نضغط زر الفأرة الايمن و نختار Insert و نختار منها Dialog ثم نضغط زر new ثم OK . أتبع الصور التالية :

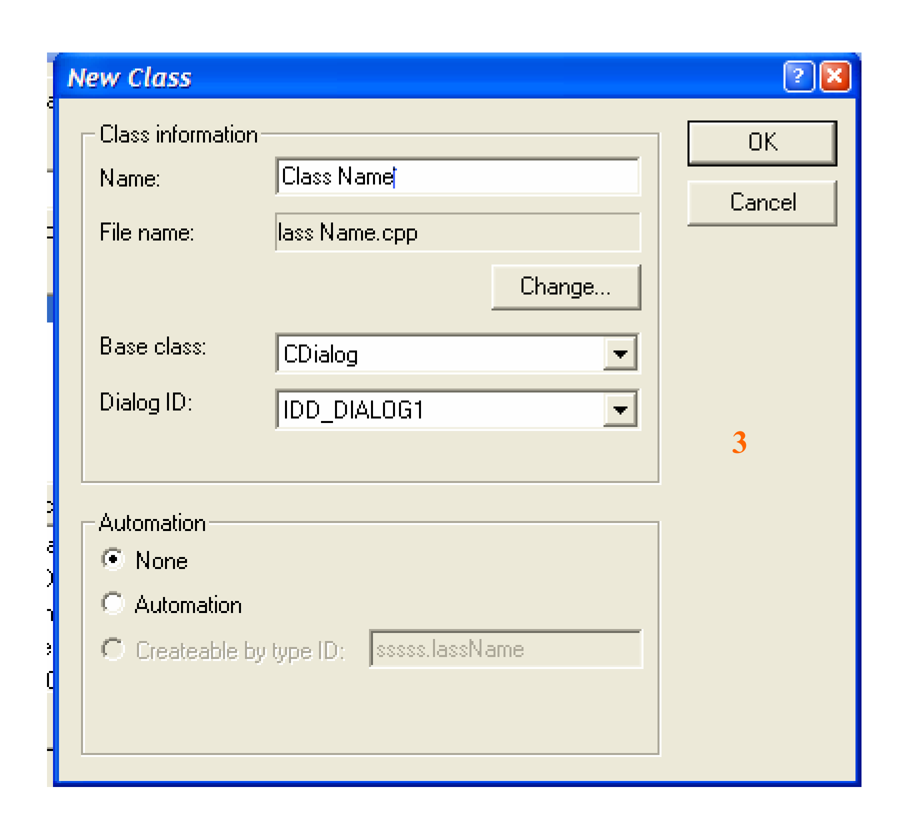




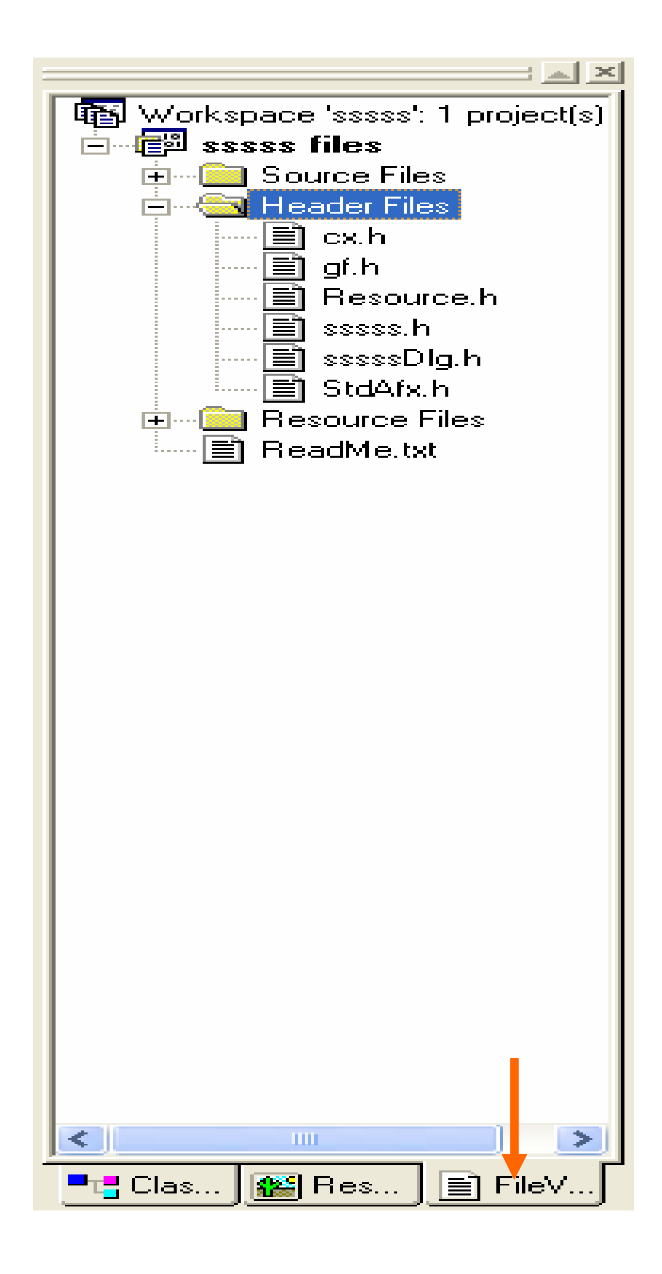
2. من View نختار Class Wizard ومنها نضغط زر OK ثم نضع اسمه ثم نضغط OK وبعدها نضغط OK أتبع الصور التالية :







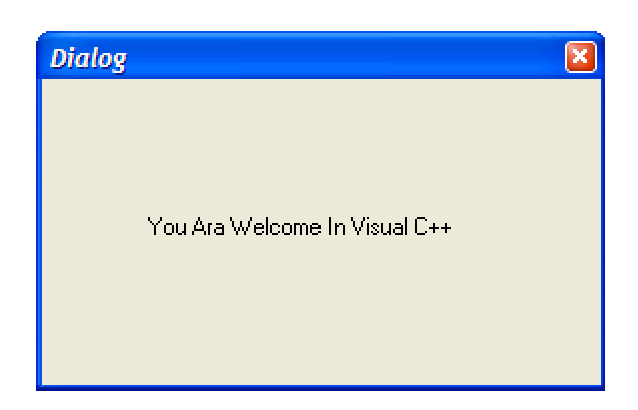
3. نذهب الى File ومنها Header Files ومنها
(اسم المشروع.h) بنفتحه و نكتب فيه "#include "ClassName.h
إتبع الصورة التالية :



مثال : قم بعمل البرنامج التالي :



عمل البرنامج
يتم التأكد من اسم المستخدم و كلمة المرور غير فارغة و اذا كانت صحيحة يتم الدخول الى الفورم الثاني و غير هيك يتم أعطاء رسالة بأن كل من اسم المستخدم أو كلمة المرور غير صحيحة .



```
void CNewDialogDlg::OnButton1()
{
	UpdateData(true);
	if (A1=="" || A2=="")
	{
		MessageBox("Pleae Enter The UserName OR Password");
	}

	if (A1=="Ayah" && A2=="1992")
	{
		Admin A;
		A.DoModal();
	}

	else
	{
		MessageBox("Please Enter Correct User OR Pasword");
	}
}
```

* **Type of Message**

أنواع الرسائل

نسطيع أن نتحكم بصندوق الرسالة من حيث :

1. اسم عنوان الرسالة
2. الخيارات التي تظهر للمستخدم
(Yes,No,Cancel,Ok, Abort,Retry,Ignore)
3. دائماً في صندوق الرسائل ترجع قيمة و هي رقم اختيار المستخدم

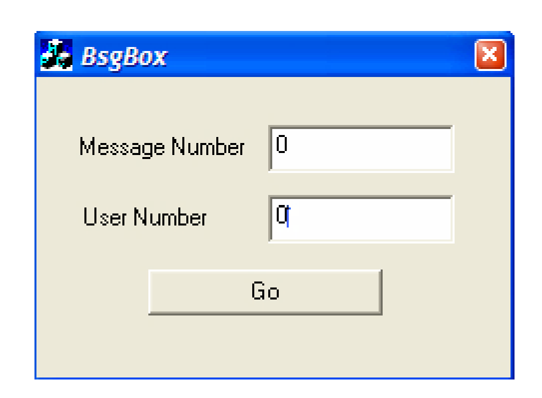
إختيار المستخدم		أرقام الرسالة	
1	OK	0	OK
2	Cancel	1	Ok + Cancel
3	Abort	2	Abort + Retry + Ignore
4	Retry	3	Yes + No + Cancel
5	Ignore	4	Yes + No
6	Yes	5	Retry + Cancel
7	No		

** أشكال الأيقونات

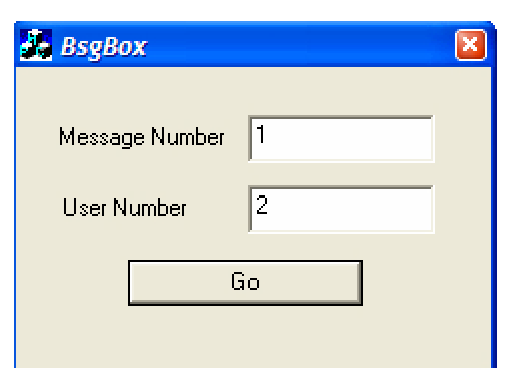
1. شكل X (رمز Stop) رقمها 16
2. شكل ? (رمز علامة الاستفهام) رقمها 32
3. شكل ! (رمز علامة التعجب) رقمها 48

** حتى يتم عرض إختيار المستخدم يجب أن نضع كود الرسالة في متغير من نوع int .

مثال : قم بعمل البرنامج التالي



عمل البرنامج
يقوم المستخدم بإدخال رقم الرسالة في الصندوق الاول و عند الضغط على زر أمر Go يتم إظهار رسالة بعنوان VC++ و نصها ...First Exam بحيث يظهر في الصندوق الثاني رقم اختيار المستخدم لأي زر تم الضغط عليه مثلاً لو قام المستخدم بإدخال رقم الرسالة 1 فإذا اختار المستخدم OK يظهر في الصندوق الثاني رقم 1 و إذا اختار Cancel يظهر رقم 2 .
كما في الشكل التالي




```
void CBsgBoxDlg::OnButton1()
{
	int a;
	UpdateData(true);
	a=MessageBox("First Exam....","VC++",A1);
	A2=a;
	UpdateData(false);

}
```

لإظهار شكل أيقونه للرسالة يتم عمل ذلك كما يلي

```
a=MessageBox("First Exam....","VC++",A1+32);
```

* Single Document Interface (SDI)

واجهة الوثيقة الواحدة

- الفرق بين Dialog based & SDI
- الفرق بين SDI & MDI

** Dialog based

يحتوي على Form و أدوات تمكننا من التصميم

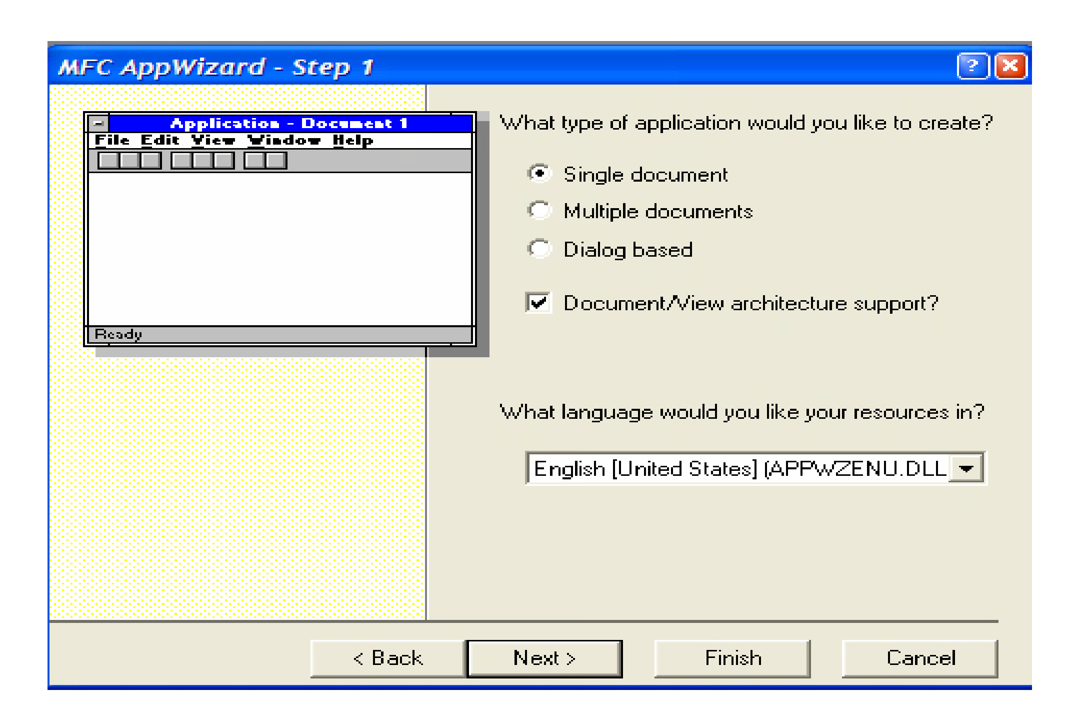
** SDI

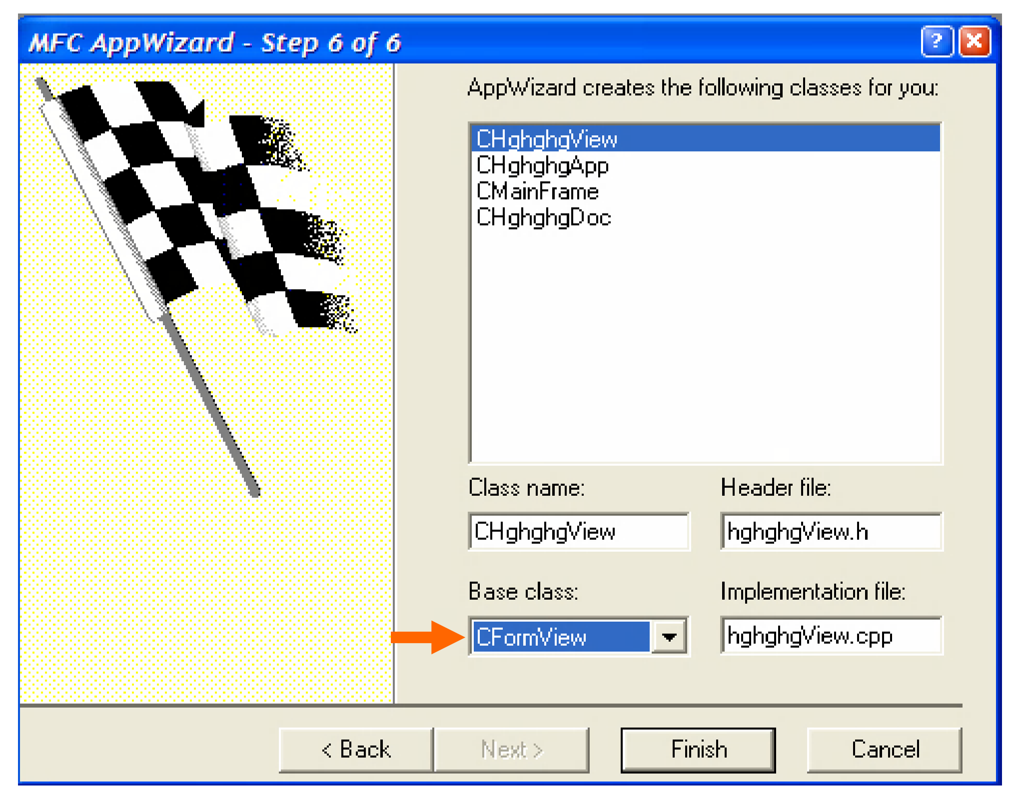
يشبه الـ NotePad و تطبيق يحتوي على قوائم و لا نستطيع فتح أكثر من وثيقة واحدة في نفس الوقت .

** MDI

يشبه الـ NotePad و تطبيق يحتوي على قوائم و نستطيع فتح أكثر من وثيقة واحدة في نفس الوقت .

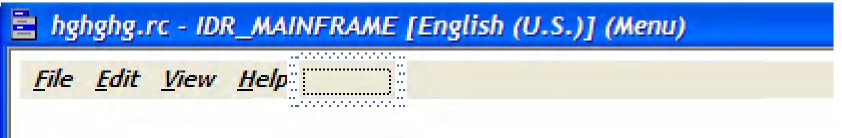
لفتح مشروع من نوع SDI نختار Single Document كما في الصور التالية و نتابع الضغط على زر Next حتى نصل زر Finish و نختار أمر Cform View ثم نضغط Finish تابع الصور التالية :





ثم نضغط على زر موافق فتظهر الشاشة الرئيسية للبرنامج كما في الشكل التالي :

** لإضافة قائمة جديدة للبرنامج نختار من Resource مجلد Menu ومنها ننقر نقر مزدوج على القائمة من المجلد
و حتى نضيف القائمة الجديدة نضغط نقر مزدوج على أخر قائمة فارغة و نضع اسم القائمة .
لاحظ الشكل التالي :



** إذا أردنا جعل قائمة تعمل عمل قائمة أخرى مثل قائمة Open نضغط على القائمة الجديدة نقر مزدوج و نجعل ID الخاص بها نفس ID الخاص بقائمة Open .

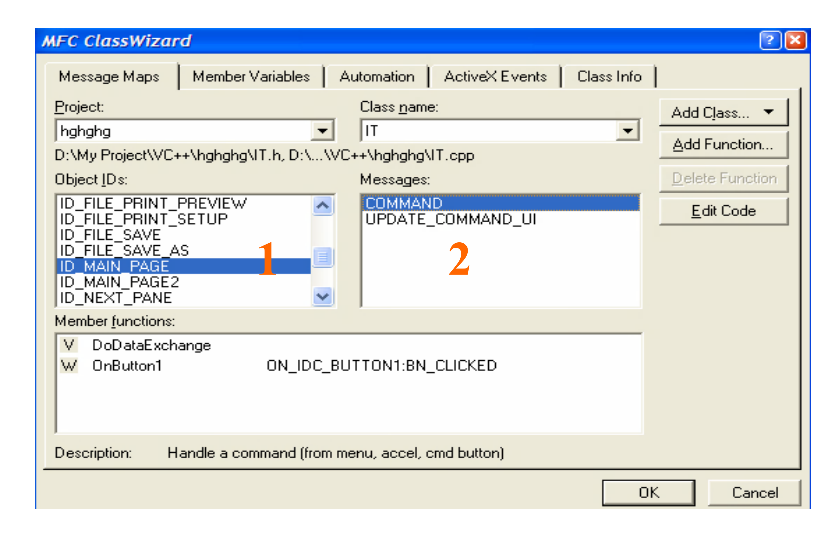
** خصائص القوائم

1. ID : لتحديد وظيفة عنصر القائمة
2. Pop-Up : لإنشاء قائمة فرعية من قائمة سابقة
3. Checked : إظهار إشارة صح بجانب اسم القائمة
4. inactive : لتعطيل العنصر في القائمة
5. Prompt : شرح بسيط عن القائمة يظهر في شريط الحالة

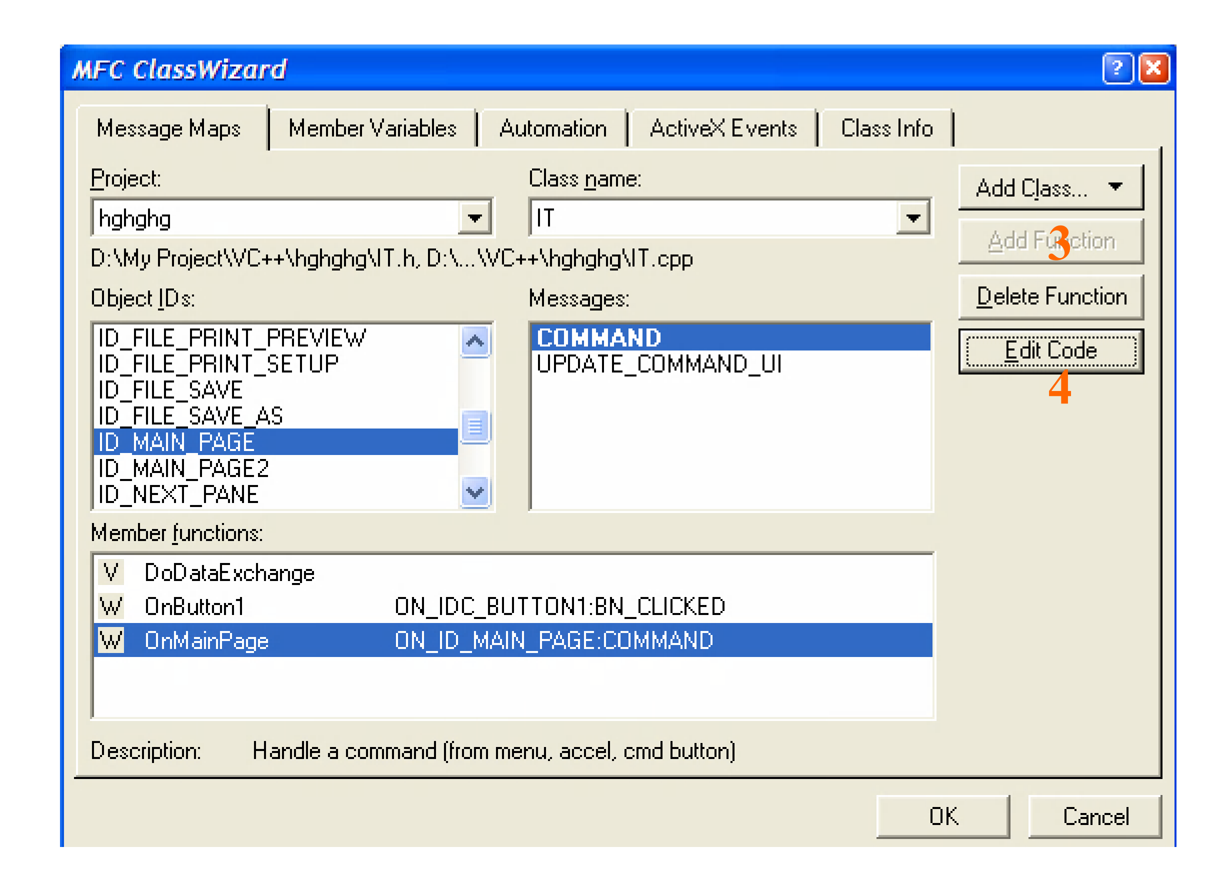
** التعامل مع القوائم

1. إضافة قائمة جديدة مع تحديد وظيفتها
2. حذف قائمة موجودة : تحديد القائمة بالفأرة و الضغط على زر Del
3. نقل مكان قائمة : تحديد القائمة بالفأرة و الاستمرار بالضغط مع السحب
4. تعديل قائمة موجودة : أي تغيير اسم القائمة \ تغيير الوظيفة عن طريق ID

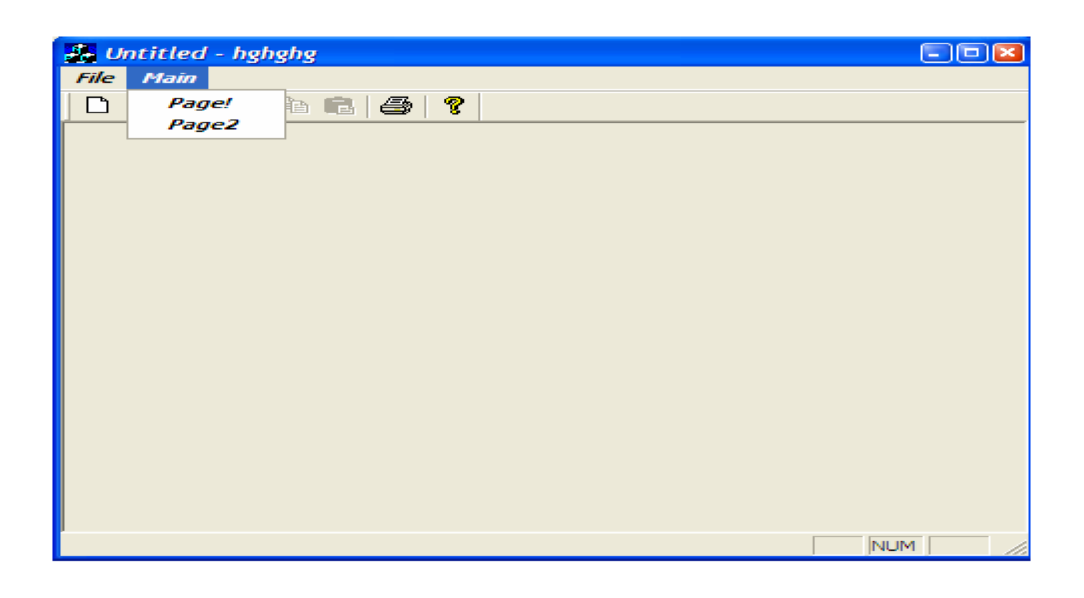
** برمجة القوائم
بعد تصميم القوائم نقوم ببرمجتها و ذلك يتم كما يلي تابع الصور من قائمة View نختار Class Wizard و منها نختار ID الخاص بالقائمة ثم نضغط على خيار Command و من ثم نضغط زر Add Function ثم نضغط OK ثم نضغط زر Edit Code فتأتي صفحة الكود لبرمجة القوائم



بعد الضغط على زر Add Function و بعدها الضغط على زر OK نضغط زر Edit Code .



مثال : قم بعمل البرنامج ملف قوائم بحيث عند الضغط على أمر في القائمة يعمل على فتح الفورم الخاص به . علماً أن البرنامج يحتوي على 2 Forms بالإضافة الى الفورم الرئيسي .



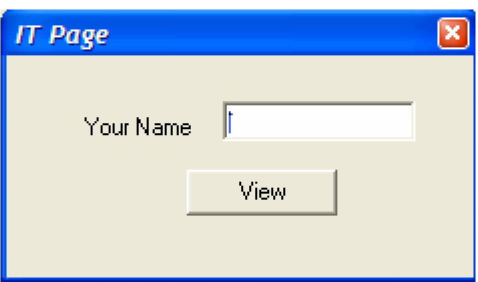
برمجة قائمة Page1 :

```
void CHghghgView::OnMainPage()
{
      IT i;
      i.DoModal();
}
```

برمجة قائمة Page2:

```
void CHghghgView::OnMainPage2()
{
      NT n;
      n.DoModal();
}
```

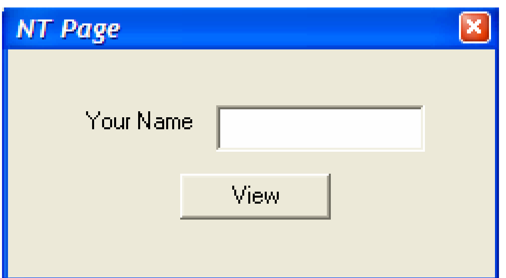
عند الضغط على قائمة Page1 يتم إظهار الفورم الثاني كما في الشكل التالي



بحيث عند الضغط على زر View يتم عرض الاسم المدخل في صندوف النص برسالة مع أيقونة إشارة التعجب .

```
void IT::OnButton1()
{
	UpdateData(true);
	MessageBox("Your Name is  " + A1+48);
	UpdateData(false);
}
```

و عند الضغط على قائمة Page2 يتم إظهار الفورم الثالث كما في الشكل التالي



بحيث عند الضغط على زر View يتم عرض الاسم المدخل في صندوف النص برسالة مع أيقونة إشارة الاستفهام .

```
void NT::OnButton1()
{
	UpdateData(true);
	MessageBox("Your Name is  "+A1+32)
	UpdateData(false);
}
```

* **Working with Mouse Event**

العمل مع أحداث الفأرة

وهي وقوع حدث معين خلال تنفيذ البرنامج مثل تحريك الفأرة أو الضغط على زر الفأرة الايمن أو الايسر ، و نستطيع الاستفادة من هذا الحدث لتنفيذ وظيفة معينة .

** أحداث الفأرة

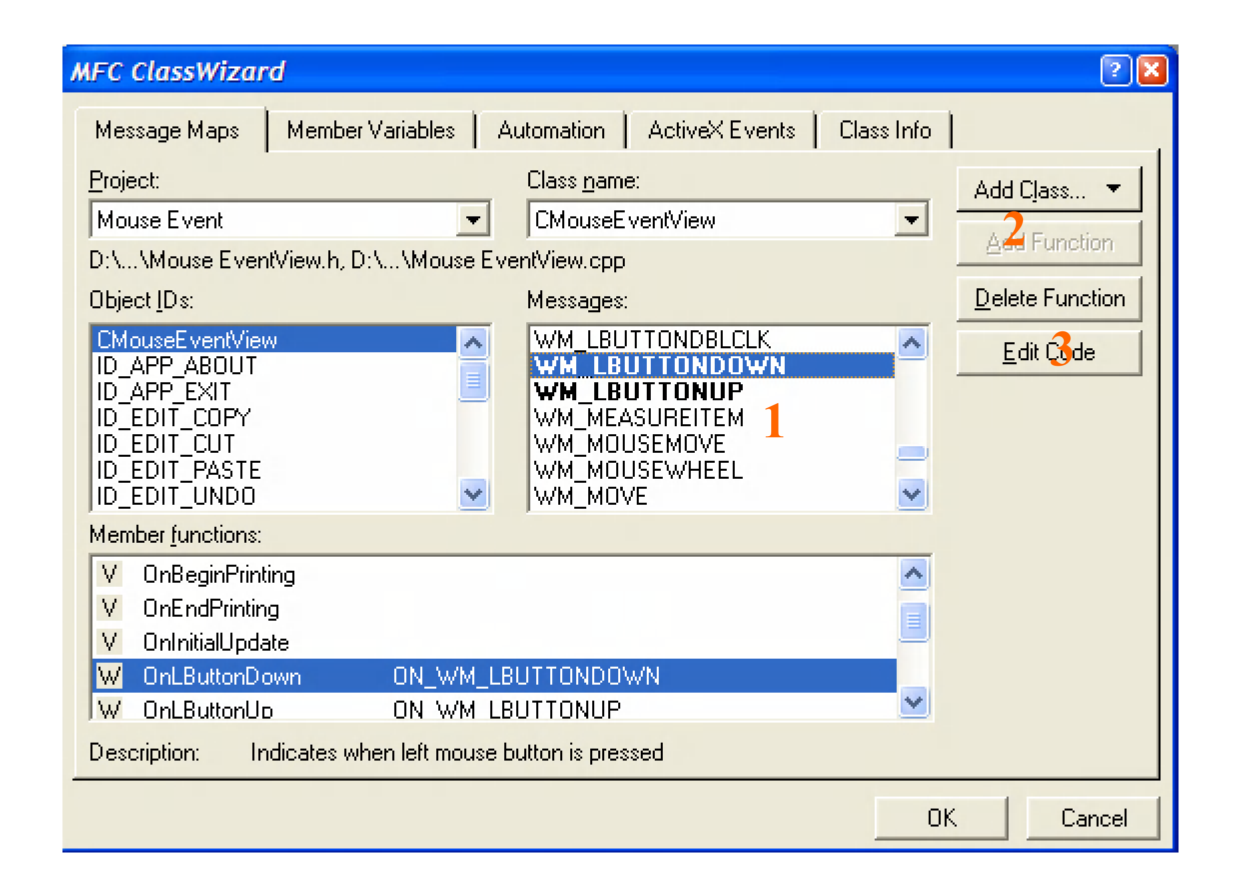
1. WM_LBUTTONDBLCLK
2. WM_LBUTTONDOWN
3. WM_LBUTTONUP
4. WM_RBUTTONDBLCK
5. WM_RBUTTONDOWN
6. WM_RBUTTONUP
7. WM_MOUSEMOVE

* كيفية إضافة كود لإحداث الفأرة

يتم ذلك عن طريق ما يلي :

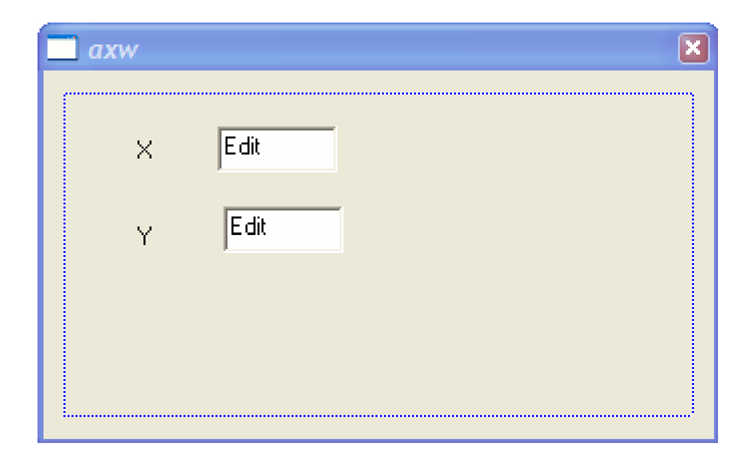
1. من قائمة View نختار Class Wizard
2. ثم نختار اسم الحدث
3. ثم نضغط على زر Add Function
4. ثم نضغط على زر Edit Code .

كما في شكل التالي :



بعدها يتم الانتقال الى صفحة الكود الخاص بالحدث ليتم برمجة الحدث

* مثال : قم بعمل برنامج يقوم بعرض الاحداث السيني في صندوق النص الاول و الاحداث الصادي في صندوق النص الثاني كما في الشكل التالي



```
void CAxwDlg::OnMouseMove(UINT nFlags, CPoint point)
{
	A1=point.x;
	A2=point.y;
	UpdateData(false);
	CDialog::OnMouseMove(nFlags, point);

}
```

** Cclint DC

و هي كلاس تستخدم للرسم تحتوي على جميع الدوال المطلوبة لرسم اشكال جاهزة و نستطيع من خلالها رسم نقطة ، خط ، مستطيل ، دائرة ، الخ ....

* كيفية استدعاء الكلاس Cclint DC
يتم استدعاء الكلاس عن طريق الجملة البرمجية التالية

Cclint Dc dc(this);

* إستخداماته :

1. لرسم نقطة dc.SetPixel
2. لرسم خط dc.LineTo
3. لرسم مستطيل dc.Rect

** الألوان الأساسية

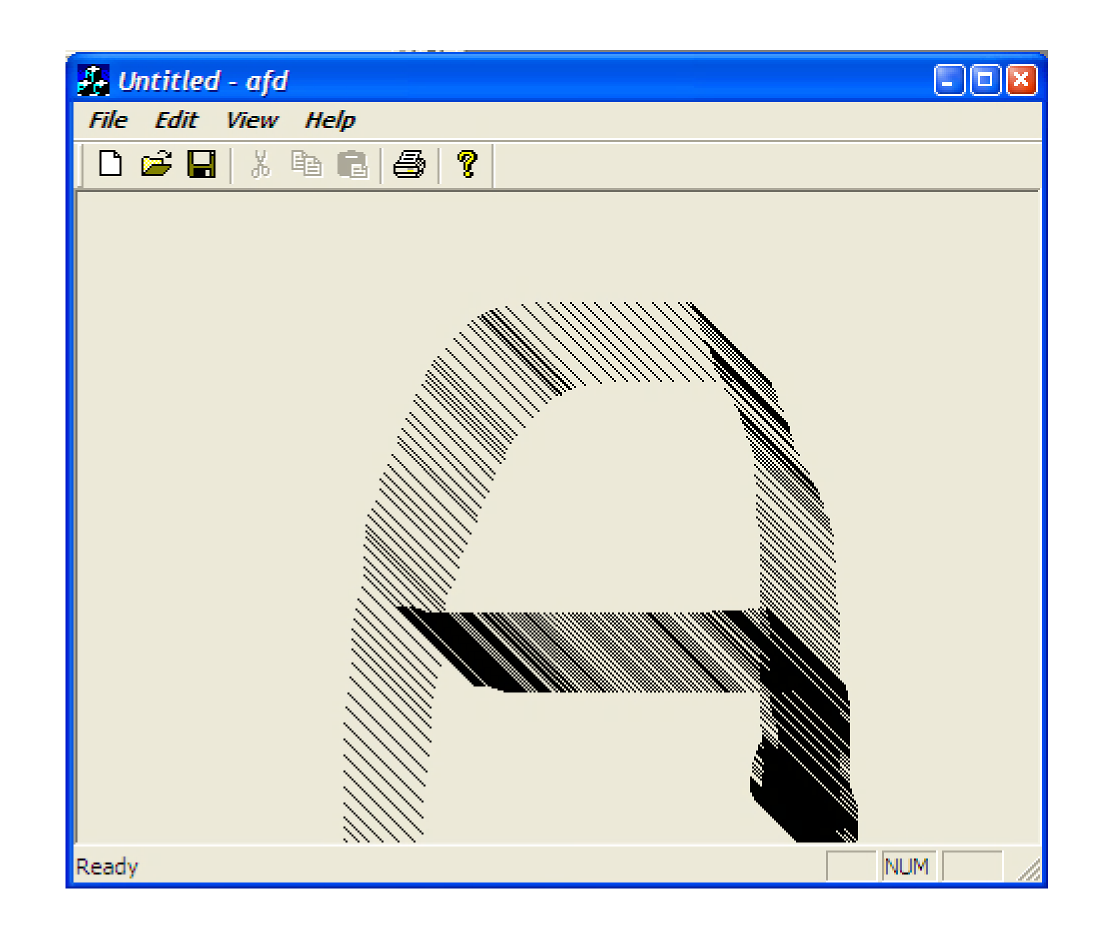
R	G	B	
255	0	0	أحمر
0	255	0	أخضر
0	0	255	أزرق
255	255	255	أبيض
0	0	0	أسود

** لرسم الخط
dc.MoveTo(20,40) لرسم نقطة البداية
dc.LineTo(100,100) لرسم نقطة النهاية

*nflags : متغير يخص الفأرة يأخذ قيمة True إذا ضغط المستخدم على زر الفأرة الأيمن أو الأيسر أو قام بتحريك الفأرة .

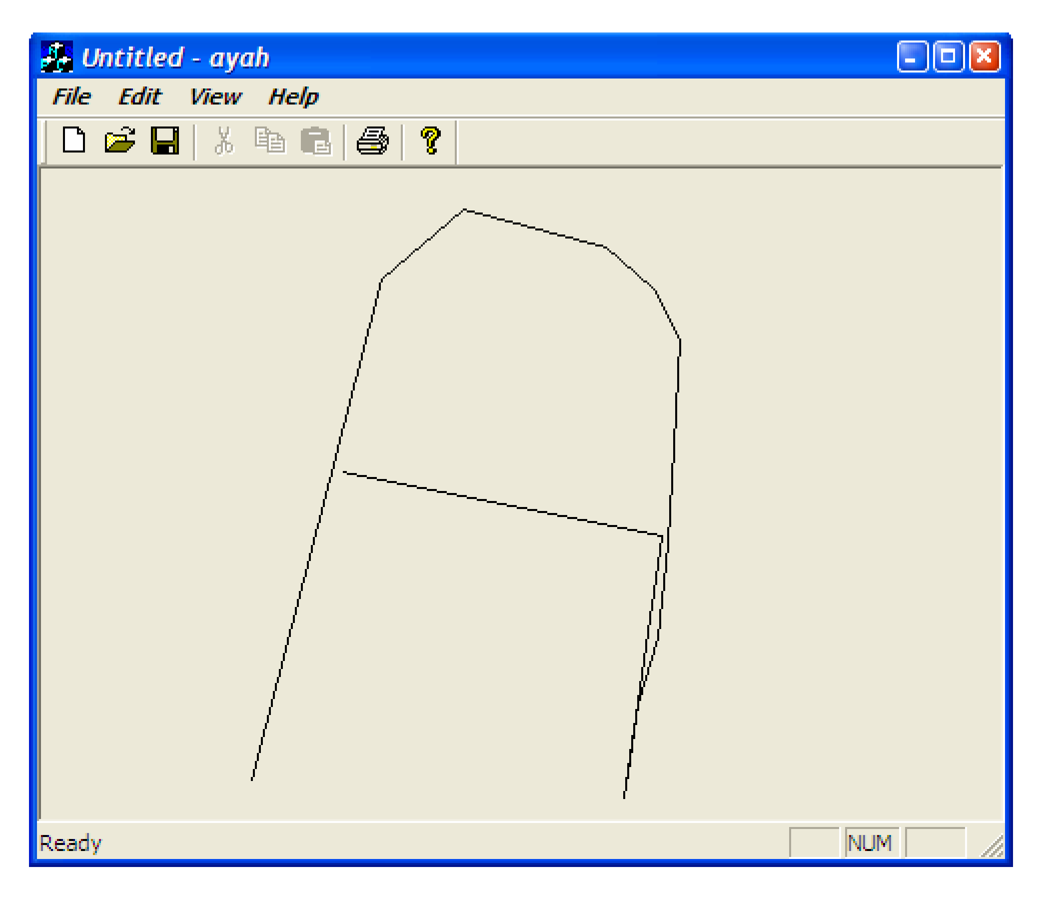
**** ملاحظة : لتعريف متغييرات مرئية لكل المشروع قم بعمل ما يلي**
FileView --- Header File --- اسم المشروعView.h--- اسفل Public Attributes
نقوم بتعريف المتغييرات

* مثال : قم بعمل برنامج يعمل على رسم خط بحيث يأخذ الاحداثيات من حركة الفأرة كما في الشكل التالي .



```
void CAfdView::OnMouseMove(UINT nFlags, CPoint point)
{
	CClientDC A(this);
	A.MoveTo(point.x,point.y);
	A.LineTo(point.x+40,point.y+40);
	CFormView::OnMouseMove(nFlags, point);
}
```

* مثال : قم بعمل برنامج يعمل على رسم خط بحيث يستطيع المستخدم تحديد نقطة البداية و النهاية عند الضغط على زر الفأرة الأيسر . كما في الشكل التالي




```
void CAyahView::OnLButtonDown(UINT nFlags, CPoint point)
{
	CClientDC A(this);
	A.MoveTo(pervX,pervY);
	A.LineTo(point.x,point.y);
	pervX=point.x;
	pervY=point.y;
	CFormView::OnLButtonDown(nFlags, point);
}
```

* تم تعريف متغييرات عامة مرئية لكل المشروع (pervX , pervY) .

* Working with Keyboard Events:

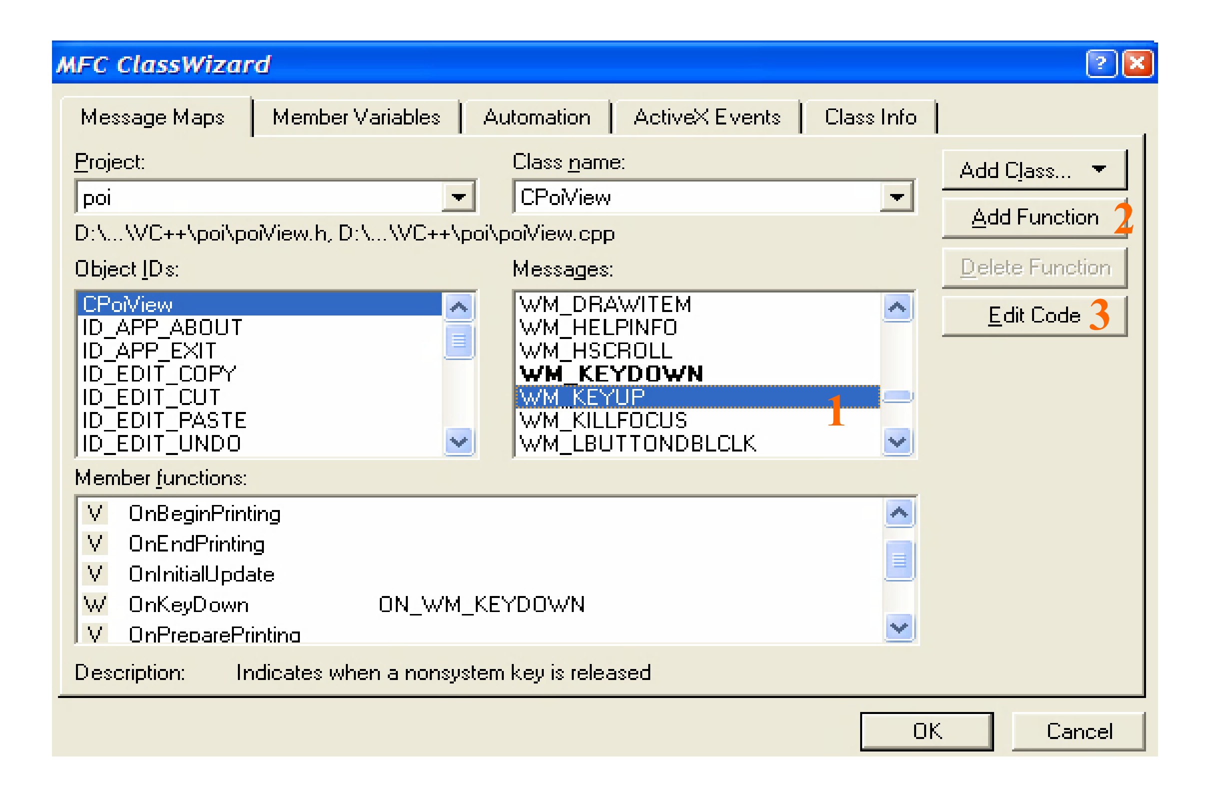
العمل مع أحداث لوحة المفاتيح

* أحداث لوحة المفاتيح
1. WM_KEYDOWN خلال الضغط على المفتاح في لوحة المفاتيح
2. WM_KEYUP إفلات المفتاح في لوحة المفاتيح
3. WM_CHAR قيمة الحرف في الـ ASCII Code

* nChar: متغير يستخدم لتخزين قيمة الحرف

** كيفية برمجة أحداث لوحة المفاتيح
يتم ذلك من خلال ما يلي :

1. الذهاب الى قائمة View و منها نختار Class Wizard
2. نختار الحدث المراد برمجته من قائمة Messages
3. الضغط على زر أمر Add Function ثم الضغط على زر أمر Edit Code فيتم الانتقال الى صفحة الكود لكتابة التعليمات البرمجية اللازمة كما في الشكل التالي :



* مثال : قم بعمل برنامج بحيث يتم برمجة أربعة مفاتيح (الاسهم) عند الضغط على أي مفتاح يتم عرض رسالة بأنه تم الضغط مفتاح .....

```
void CPoiView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
 // TODO: Add your message handler code here and/or call default

 if (nChar==VK_UP)
      MessageBox("Your Pressed Up Row Key");
 if (nChar==VK_DOWN)
      MessageBox("Your Pressed Down Row Key");
 if (nChar==VK_LEFT)
      MessageBox("Your Pressed Left Row Key");
 if (nChar==VK_RIGHT)
      MessageBox("Your Pressed Right Row Key");
 CFormView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```

** Graphics

** الرسم

* ورقة الرسم تتوفر في بيئة ++VC من خلال الفصيلة (الكلاس) Device Context

* Device Context : و هي فصيلة (كلاس) جاهزة توفر البيئة اللازمة و الدوال الجاهزة لرسم أشكال معينة مثل مستطيل ، دائرة ، شكل بيضوي.

* كيفية إستدعاء الفصيلة (الكلاس) بحيث يرمز لها بالرمز CDC يتم إستدعائها كما يلي :

CDC * P

- CDC :الكلاس
- * : عبارة عن مؤشر
- P :عبارة عن كائن من الكلاس

* المؤشر (*) : يعني في الذاكرة الرئيسية بيكون مخزن ألاف من الدوال و الإقترانات ، حتى لا يحدث جهد على البرنامج و يسهل عمله يتم نقلهم الى الذاكرة الفرعية و يكون * عبارة عن مؤشر في الذاكرة الرئيسية الى الذاكرة الفرعية .

* لرسم شكل مستطيل

```
void CXcvView::OnButton2()
{
     CDC * A = GetDC();
     A->Rectangle(40,50,90,90);
}
```

* لرسم شكل خط

```
void CXcvView::OnButton3()
{
     CDC * A = GetDC();
     A->MoveTo(70,40);
     A->LineTo(100,60);
}
```

* لرسم شكل بيضوي

```
void CXcvView::OnButton6()
{
     CDC * A= GetDC();
     A->Ellipse(60,40,180,140);
}
```

* لرسم شكل قوس

```
void CXcvView::OnButton4()
{
     CDC * A = GetDC();
     A->Arc(40,50,90,90,60,60,80,90);
}
```

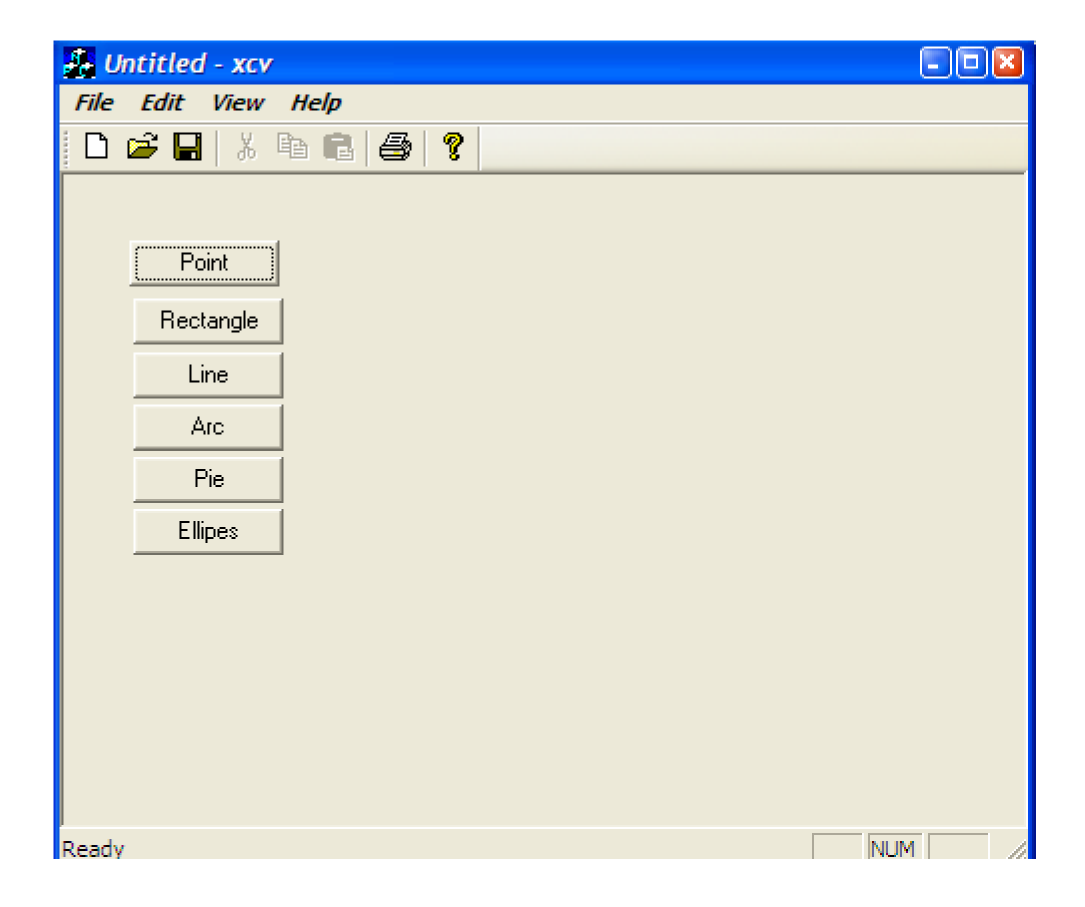
* لرسم شكل مخروط

```
void CXcvView::OnButton5()
{
     CDC * A = GetDC();
     A->Pie(40,60,70,100,50,50,70,90);
}
```

* لرسم نقطة

```
void CXcvView::OnButton1()
{
	CDC * A = GetDC();
	A->SetPixel(20,30,RGB(0,0,0));
}
```

يتم وضع هذه الأسطر البرمجية داخل أزرار أمر كما في الشكل التالي :



* Graphics Pin

**** القلم**

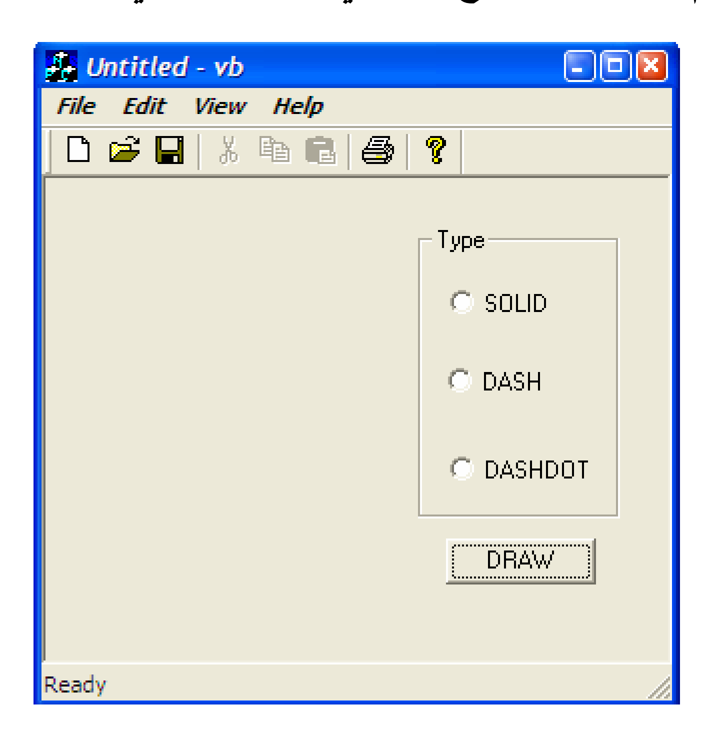
حيث أن القلم يتعامل مع ثلاث معطيات و هي :

1. اللون : يتعامل مع نظام RGB .
2. الحجم : 255 – 0
3. النوع : / PS_SOLID / PS_DASH
PS_DASHDOT / PS_NULL

*** كيفية إستدعاء فصيلة (كلاس) القلم**
يتم ذلك من خلال

Cpen P (type, size, color);

*** مثال : قم بعمل البرنامج كما في الشكل التالي :**



بحيث يتم رسم شكل مستطيل بناءً على نوع القلم و باللون الاسود

```
void CVbView::OnButton1()
{
    if (A==0)
    {
        CDC * P=GetDC();
        CPen Pn(PS_SOLID,1,RGB(0,0,0));
        P->SelectObject(Pn);
        P->Rectangle(100,100,200,200);
    }
    if (A==1)
    {
        CDC * P=GetDC();
        CPen Pn(PS_DASH,1,RGB(0,0,0));
        P->SelectObject(Pn);
        P->Rectangle(100,100,200,200);
    }
    if (A==2)
    {
        CDC * P=GetDC();
        CPen Pn(PS_DASHDOT,1,RGB(0,0,0));
        P->SelectObject(Pn);
        P->Rectangle(100,100,200,200);
    }
}
```

```
* P->SelectObject(Pn);
```

و تعني يتم أخذ خصائص الكائن **Pn** و يتم تطبيقها على الكائن **P** .

* Graphics Font

** الخط

* للتعامل مع الخط هنالك كلاسين من خلالهما يمكن التعامل مع الخط هما :

1. الطريقة المباشرة Create Font : يجب تحديد جميع خصائص الخط .
2. الطريقة غير المباشرة Create Font Indirect : و هي الطريقة الاسهل حيث يتم تحديد ما تريد من خصائص الخط فقط .

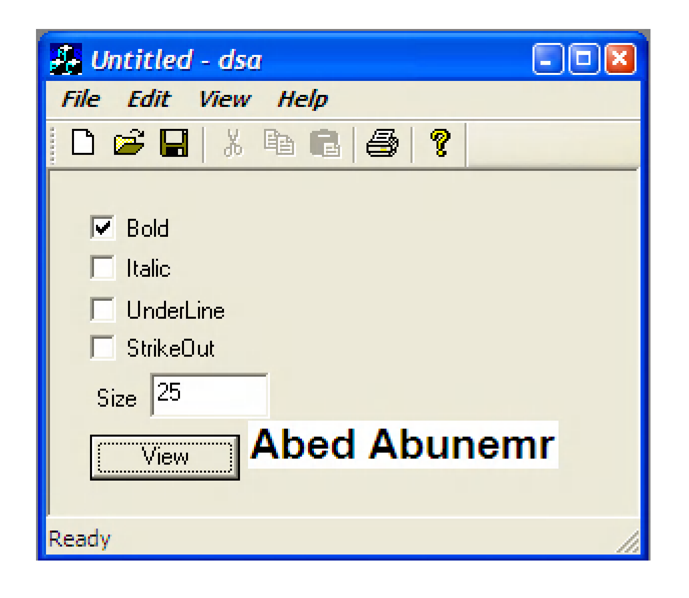
*** الطريقة المباشرة Create Font**
تتم بالطريقة التالية

```
CFont F;
F.CreateFont(Height,Width,Escapement,Orientation,Weight
,Italic,Underline,StrikeOut,CharSet,OutPrecision,ClipPr
ecision,Quality,PitchAndFamily,Facename);
```

*** الطريقة غير المباشرة Create Font Indirect**
تتم الطريقة التالية

```
CFont A;
static LOGFONT lf;
lf.lfHeight = 20;
lf.lfWidth = 10;
lf.lfItalic = true;
A.CreateFontIndirect(& lf);
```

*** مثال قم بعمل برنامج يقوم البرنامج عند تفعيل أي من خصائص و تحديد حجم الخط و الضغط على زر View يتم الكتابة على الشاشة بالمواصفات التي تم تحديدها . كما في الشكل التالي :**




```
void CDsaView::OnButton1()
{
	UpdateData(true);
	if (a1==true)
	{
		CFont A;
		static LOGFONT lf;
		lf.lfWeight = 700;
		lf.lfHeight= A1;
		A.CreateFontIndirect( & lf);
		CDC * P = GetDC();
		P->SelectObject(A);
		P->TextOut(100,125,"Abed Abunemr");
	}
	if (a2==true)
	{
		CFont A;
		static LOGFONT lf;
		lf.lfItalic= true;
		lf.lfHeight= A1;
		A.CreateFontIndirect( & lf);
		CDC * P = GetDC();
		P->SelectObject(A);
		P->TextOut(100,125,"Abed Abunemr");
	}
	if (a3==true)
	{
		CFont A;
		static LOGFONT lf;
		lf.lfUnderline= true;
		lf.lfHeight= A1;
		A.CreateFontIndirect( & lf);
		CDC * P = GetDC();
		P->SelectObject(A);
		P->TextOut(100,125,"Abed Abunemr");
	}
```

```
        if (a4==true)
        {
              CFont A;
              static LOGFONT lf;
              lf.lfStrikeOut=true;
              lf.lfHeight= A1;
              A.CreateFontIndirect( & lf);
              CDC * P = GetDC();
              P->SelectObject(A);
              P->TextOut(100,125,"Abed Abunemr");
        }
}
```

*ملاحظات :

- لجعل النص بالنمط العريض Bold يجب إعطائه قيمة 700
- لجعل النص بالنمط العادي UnBold يجب إعطائه قيمة 400
- التعليمة التالية `A.CreateFontIndirect( & lf);` حيث أن إشارة & تعني دمج جميع خصائص الخط

** Timer

* Timer : عبارة عن مؤقت (يرسل نبضة الى البرنامج) نستطيع فيه تحديد الوقت و تحديد العملية المطلوب تنفيذها و كذلك نستطيع إيقاف الـــ Timer .

* كيفية إنشاء Timer
يتم ذلك من خلال التعليمة التالية :

```
Set Timer (ID, Time, Operation);
```

حيث أن :

1. ID : رقم الـــ Timer
2. Time : الوقت الذي ستنفذه به العملية علماً أن (0.5 ثانية = 500 ميلي ثانية) و (1 ثانية = 1000 ميلي ثانية)
3. Operation: العملية المطلوب تنفيذها عند الوقت المحدد.

* مكان إنشاء المؤقت في حدث WM_CREATE.
* كيفية إيقاف الـــ Timer :
يتم ذلك من خلال التعليمة التالية :

```
Kill Timer (ID);
```

حيث أن ID : رقم المؤقت المراد إيقافه .

* نستطيع إستخدام المؤقت كعداد
* WM_CREATE : حدث ينفذ عند تنفيذ البرنامج بالنسبة للبرنامج من نوع SDI .

* مثال قم بعمل البرنامج التالي



يعمل البرنامج استخدام المؤقتيين كعداد يظهر في صندوق النص و عند الضغط على زر الامر الاول StopTimer1 يتم إنهاء المؤقت الاول و عند الضغط على زر الامر الثاني StopTimer2 يتم إنهاء المؤقت الثاني

** ملاحظة : تعملنا فيما سبق كيفية إضافة كود الاحداث بنفس الطريقة يتم إضافة حدث WM_TIMER && WM_CREATE للبرنامج حتى يتم برمجتهما .

```
void CTimerView::OnButton2()
{
	KillTimer(2);

}

void CTimerView::OnButton1()
{
	KillTimer(1);
}

int CTimerView::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
	if (CFormView::OnCreate(lpCreateStruct) == -1)
		return -1;

	SetTimer(1,1000,NULL);
	SetTimer(2,1000,NULL);
	return 0;
}

void CTimerView::OnTimer(UINT nIDEvent)
{
	A1++;
	UpdateData(false);
	CFormView::OnTimer(nIDEvent);
}
```

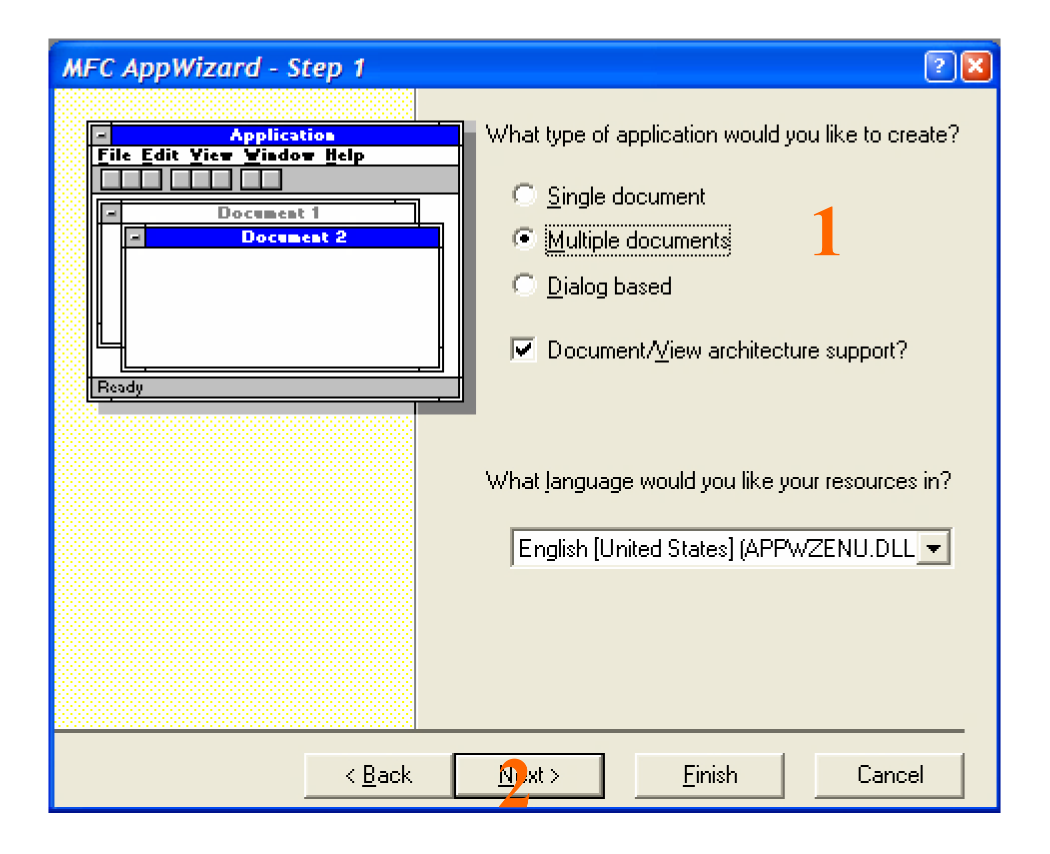
* **WM_TIMER :** إقتران خاص بالمؤقت . إذا كانت قيمة العملية NULL يقوم بتنفيذ ما في داخل WM_TIMER كل فترة . لاحظ الكود السابق .

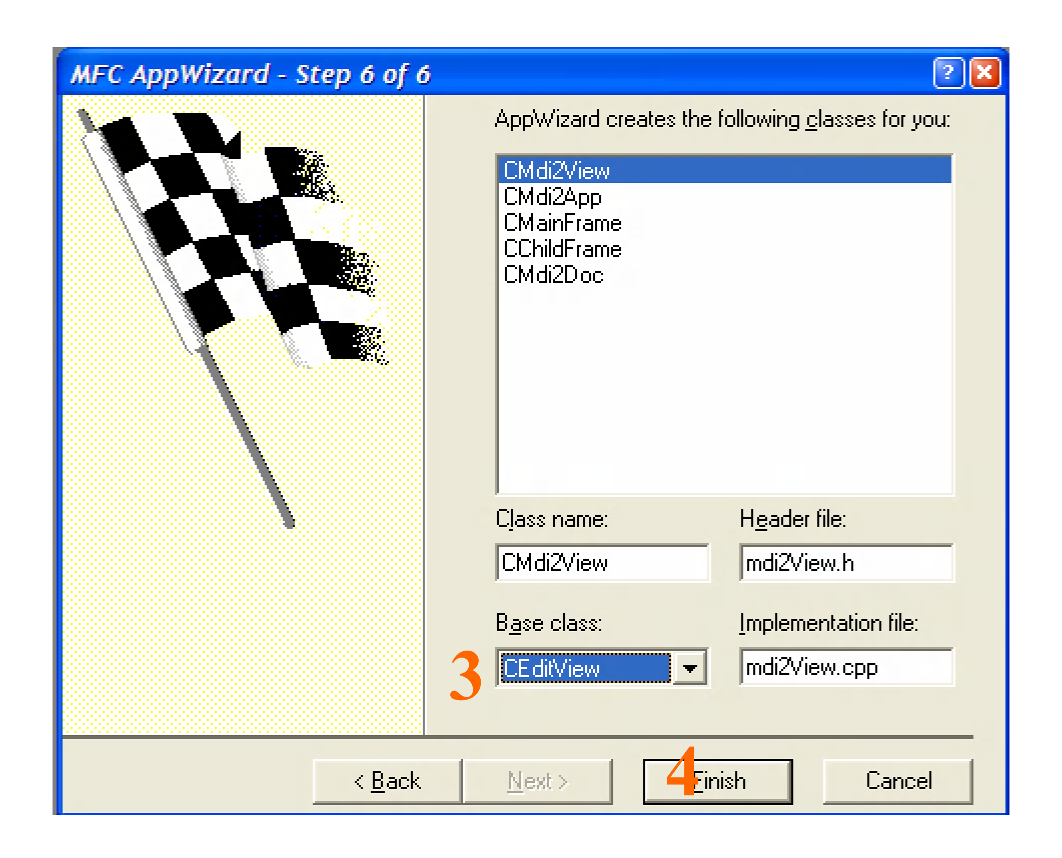
* Multi Document Interface (MDI)

واجهة الوثيقة الواحدة

وهي برنامج شبيه بإداة الكتابة Micosoft Word و هي من فصيلة (CEditeView) و نستطيع من خلالها التعامل مع نص و خصائصه و نستطيع فتح أكثر من وثيقة في نفس الوقت .
تابع الصور التالية لمعرفة كيفية فتح برنامج من نوع MDI

نختار Multi Document Interface بدلاً من dialog based ثم نتابع الضغط على زر Next حتى نصل الى زر Finish فنختار CEditView ثم نضغط على زر Finish كما في الصور التالية :





** MDI غير مستخدمة كثيراً لذا لن نتطرق لها أكثر من هكذا .

** Advanced Control

الأدوات المتقدمة

و الأدوات هي :

1. Progress Control
2. Slider Control
3. Spinner Control
4. List Control
5. Tree Control
6. Rich Edit Control

سوف يتم شرح هذه الادوات شرح وافي لكل أداة منها إنشاء الله تعالى .

* Progress Control

أداة الأستمرارية

تستخدم هذه الأداة لتعبير عن مدى إنجاز مهمه معينة و ما تبقى من الوقت اللازم لإتمام العملية .

من أشهر إستخداماتها (عملية التحميل ، عملية النقل) و هي من فصيلة (كلاس) CProgressCtrl .

* دوال هذه الاداة و هي التي سيتم إستخدامها في برمجة هذه الاداة

1. SetRange (start, End);

تستخدم لتحديد نهاية و بداية المدى الافقي للأداة

2. SetPos (int);

تستخدم لتحديد القيمة الابتدائية لموقع المؤشر في الاداة

3. GetPos ();

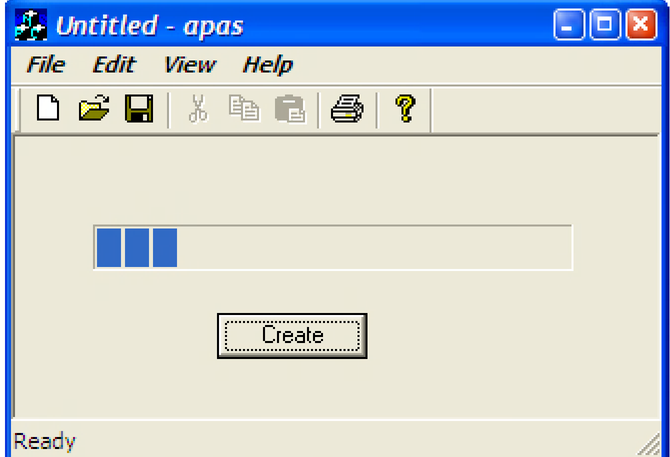
ترجع موقع المؤشر في الأداة

4. SetStep ();

لتحديد مدى كل قفزة للمؤشر في الأداة

** لأنشاء Progress Control يجب إستخدام ثلاثة دوال و هي : الدوال التي تحمل الارقام التالية كما في أعلاه : الدالة رقم 1 و رقم 2 و رقم 4 .

* قم بعمل برنامج تحميل مستخدماً أداة Progress Control كما في الشكل التالي :



خطوات عمل البرنامج

تحديد نقطة البداية و النهاية
تحديد مقدار القفزة
تحديد موقع المؤشر على الاداة

```
void CApasView::OnButton1()
{
    CProgressCtrl * A= (CProgressCtrl*)
    GetDlgItem(IDC_PROGRESS1);
    A->SetRange(80,400);
    A->SetPos(120);
    A->SetStep(40);

}
```

* <u>Slider Control</u>

شريط المسار
و هو عبارة عن مؤشر يتحرك في المدى الأفقي كل قفزة للمؤشر تعبر عن قيمة . و هو من فصيلة (كلاس) CSliderCtrl .

* دوال أداة الـ Slider Control :

1. SetRange (start, end);

لتحديد المدى الافقي للأداة

2. SetPos (int);

لتحديد موقع المؤشر في الأداة

3. SetTic (int);

لتحديد مدى القفزة الواحدة

4. GetPos ();

يرجع موقع المؤشر في الأداة

* يمكن إستخدام ما يلي بدلاً من SetRange(,);

SetRangeMax (int);

لتحديد نهاية المدى

SetRangeMin (int);

لتحديد نهاية المدى

* **Spinner**

أداة تستخدم كعداد حيث يقوم السهم العلوي بزيادة القيمة و السهم السفلي إنقاص القيمة و أكثر إستخداماته مع EditBox
و هي من فصيلة (كلاس) CPinButtonCtrl .

* دوال هذه الاداة

1. SetRange (start, end);

   لتحديد المدى الافقي للأداة (أعلى قيمة و أدنى قيمة للأداة)

2. SetPos (int);

   لتحديد القيمة التي سيبدأ بها هذه الاداة

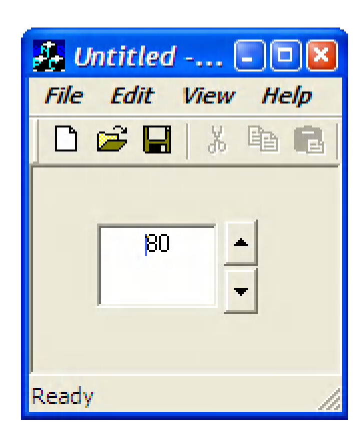
3. GetRange ();

   ترجع المدى الافقي للاداة (المدى بالكامل)

4. GetPos ();

   ترجع القيمة الحالية للاداة

* قم بعمل البرنامج التالي كما في الشكل



يعمل البرنامج عند الضغط على الاسهم يتم زيادة القيمة الموجودة في صندوق النص .

```
void CSpinnerView::OnVScroll(UINT nSBCode, UINT nPos,
CScrollBar* pScrollBar)
{
// TODO: Add your message handler code here and/or call default
     A1++;
     UpdateData(false);
     CFormView::OnVScroll(nSBCode, nPos, pScrollBar);
}
```

* يتم إضافة الكود السابق داخل حدث WM_VSCROLL
تعلمنا كيف يتم إضافة الاحداث

* ملاحظة : إذا كان الاداة بشكل أفقي يتم وضع الكود داخل حدث WM_HSCROLL و إذا كانت بشكل عمودي يتم وضع الكود داخل حدث WM_VSCROLL كما في المثال السابق

* List Control

يجب أن نأخذ منها متغير من CListCtrl .

الفرق بين الـ List Box و الـ List Control .

- List Control : تدعم عملية الصور و الرموز للعناصر و عند النقر على الصورة يتم تنفيذ أمر معين.
- List Box : لا تدعم الذي ذكر أعلاه.

* دوال هذه الاداة

1. InsertItem (columNum, Text);

لإضافة عنصر في موقع محدد

2. SetTextColor (RGB (0, 0, 0));

لتحديد لون كتابة النص

3. SetBKColor (RGB (0, 0, 0));

لتحديد لون خلفية العنصر

4. SetItemText (columNum, SubNum);

لتحديد نص للنعصر المحدد

SubNum: لتحديد رقم العنصر في العمود المحدد

5. DeleteItem (ItemNum);

لحذف العنصر المحدد من الاداة

6. DeleteAllItem ();

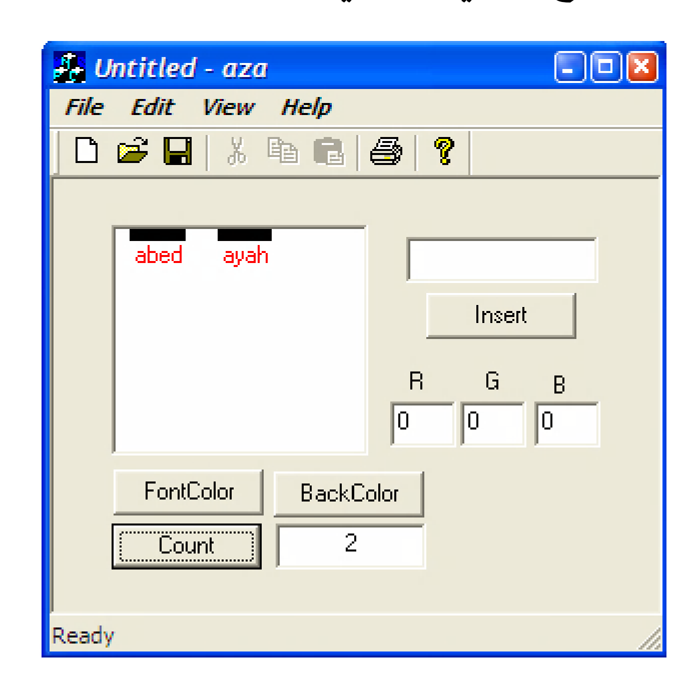
لحذف جميع العناصر من الاداة

7. GetCountPerPage ();

ترجع عدد العناصر في الاداة

* هذه الاداة من فصيلة (كلاس) CListCtrl .

* قمل بعمل البرنامج التالي كما في الشكل



عند الضغط على زر Insert يتم إضافة النص الموجود في الصندوق الاول الى القائمة .

عند الضغط على زر FontColor يتم تغيير لون الخط الى اللون المحدد في صناديق اللوان RGB

عند الضغط على زر BKColor يتم تغيير لون خلفية العنصر الى اللون المحدد في صناديق اللوان RGB

عند الضغط على زر Count يتم إظهار عدد العناصر الموجودة في القائمة في صندوق النص المجاور لزر Count

```
void CAzaView::OnButton1()
{
    UpdateData(true);
    A.InsertItem(0,A1);
    A1="";
    UpdateData(false);
}

void CAzaView::OnButton2()
{
    UpdateData(true);
    A.SetTextColor(RGB(a1,a2,a3));
    UpdateData(false);
}

void CAzaView::OnButton3()
{
    UpdateData(true);
    A.SetBkColor(RGB(a1,a2,a3));
    UpdateData(false);
}

void CAzaView::OnButton4()
{
    UpdateData(true);
    a4=A.GetCountPerPage();
    UpdateData(false);
}
```

* **Tree Control**

وهي أداة تستخدم لعرض العناصر بشكل شجرة و يوجد بها عناصر رئيسية و عناصر فرعية و أكثر إستخداماتها هي الفهارس
و هي من فصيلة (كلاس) CTreeCtrl .

* دوال هذه الاداة

1. InsertItem (columNum, text);
2. SetTextColor (RGB (0, 0, 0));
3. SetBKColor (RGB (0, 0, 0));
4. DeleteItem (index);
5. DeleteAllItem ();
6. GetCount ();

وهذه الدوال هي نفس دوال أداة List Control من حيث العمل .

* **RechEditControl**

أداة تستخدم لعمليات الكتابة تشبه Edit Box و لكن تحتوي على خصائص أكثر من حجم الخط لون الخط شكل الخط اتجاه الخط و من أشهر استخداماتها الـــ Chat .

** Toolbar && Status Bar

شريط الادوات وشريط الحالة

: Toolbar *

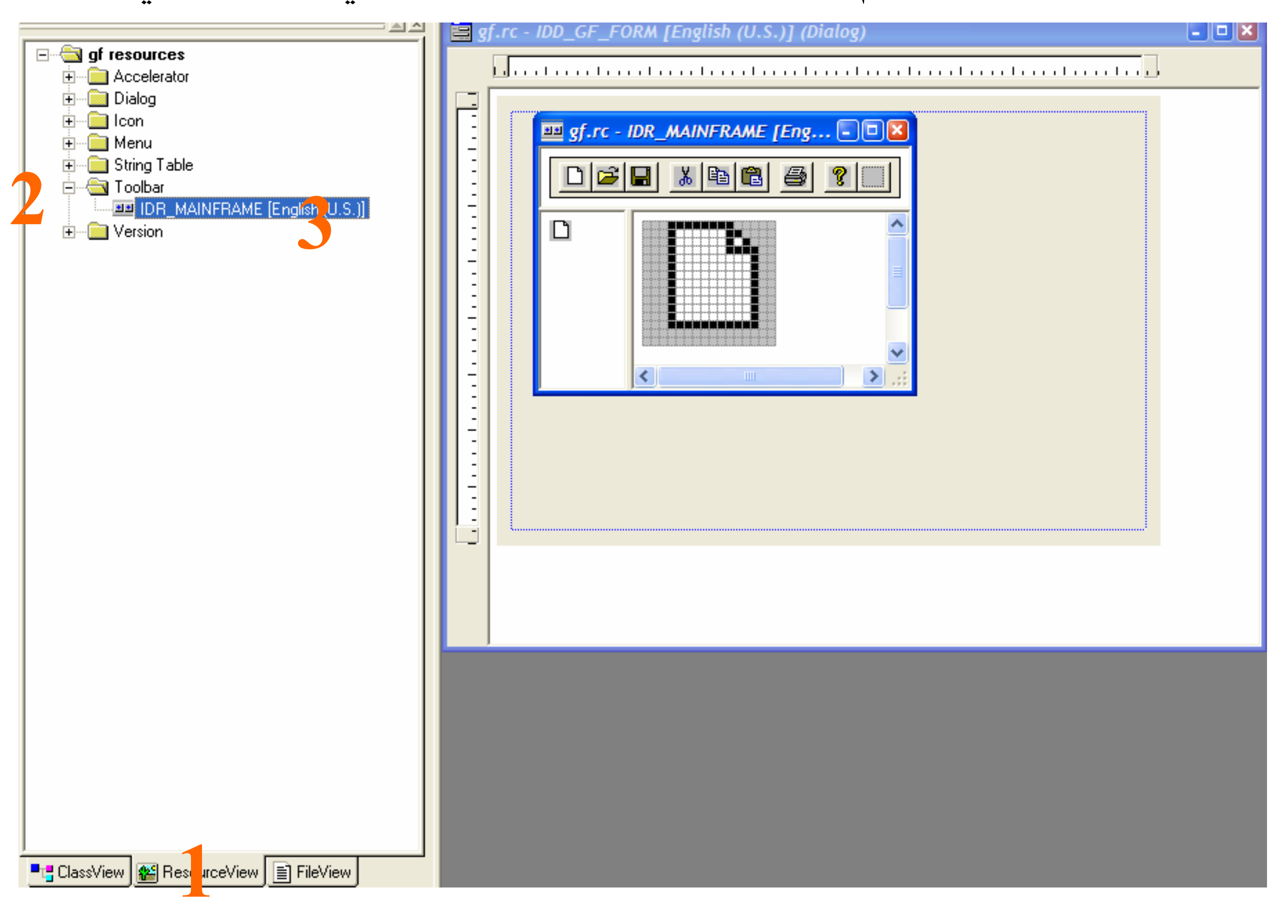
شريط موجود في أعلى الشاشة في لغة ++VC يحتوي على الادوات الاساسية أو الرئيسية وهي اختصارات شريط القوائم و يمكن حذف ، إضافة ، تعديل الاداة أو المهمة .

و هو من فصيلة (كلاس) Ccontrol وهو من كلاس Cwnd حيث أن جميع الكلاسات متفرعة من كلاس Cwnd .

* للدخول الى شريط الادوات كما يلي

ResourceView – Toolbar – IDR_MAINFRAME

نقوم بالذهاب الى ResourceView و منها نختار مجلد Toolbar ثم ننقر نقر مزدوج على IDR_MAINFRAME فتظهر شريط الادوات ليتم عليه التعديلات المطلوبة كما في الشكل التالي



* تعديل أيقونه في شريط الادوات ما هو ممكن أن يتم تعديله ، يتم تعديل ما يلي :

- تعديل الاسم : يتم ذلك كما يلي

Prompt جزء التعليق \n جزء الاسم

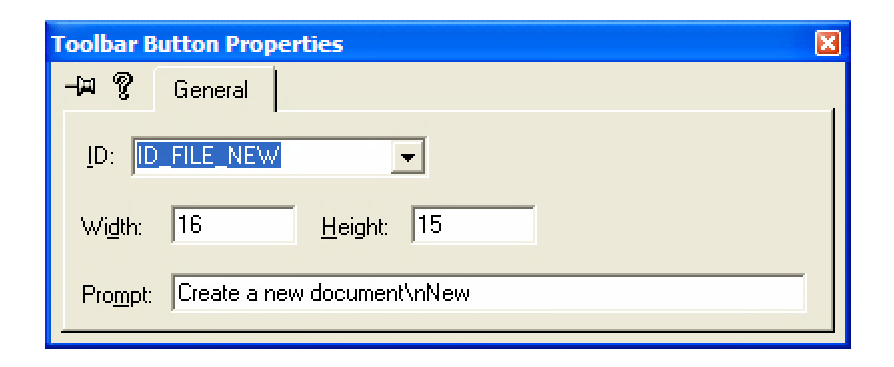
جزء الاسم يظهر عند وضع المؤشر على الاداة

جزء التعليق يظهر في شريط الحالة

- تعديل المهمة : يتم ذلك بتغيير ID لهذه الاداة

- تعديل الصورة : بوضع صورة أو رسم شكل مناسب للاداة التي تريدها في برنامجك لان الصورة تظهر في محرر يشبه الرسام .

لعمل ما ذكر أعلاه يتم النقر المزدوج على الاداة فيظهر صندوق حوار كما في الشكل التالي :



* إضافة أيقونة جديدة :

1. الضغط على نهاية شريط الادوات (الايقونة الفارغة)
2. تحديد الصورة
3. تحديد الاسم و التعليق
4. تحديد المهمة

** بالترتيب : تحديد المهمة / تحديد الاسم / تحديد الصورة

* **: Status Bar**

و هو شريط موجود في أسفل الشاشة يعطي شرح توضيحي عن الحالة في البرنامج أو شرح عن أداة و يقسم الى قسمين :

1. Message Map: شرح عن الاداة أو حالة البرنامج
2. Indicator Bar :

يحتوي على ASCII CODE للمفاتيح الاساسية في لوحة المفاتيح و يمكن الاضافة و التعديل عليها .

- Message Map :

للتعديل على الاوامر أو الشرح الذي سيظهر في هذا القسم

نذهب الى String Label و منها Caption وهو الشرح الذي يظهر في اسفل الشاشة للاداة

- Indictor Bar :

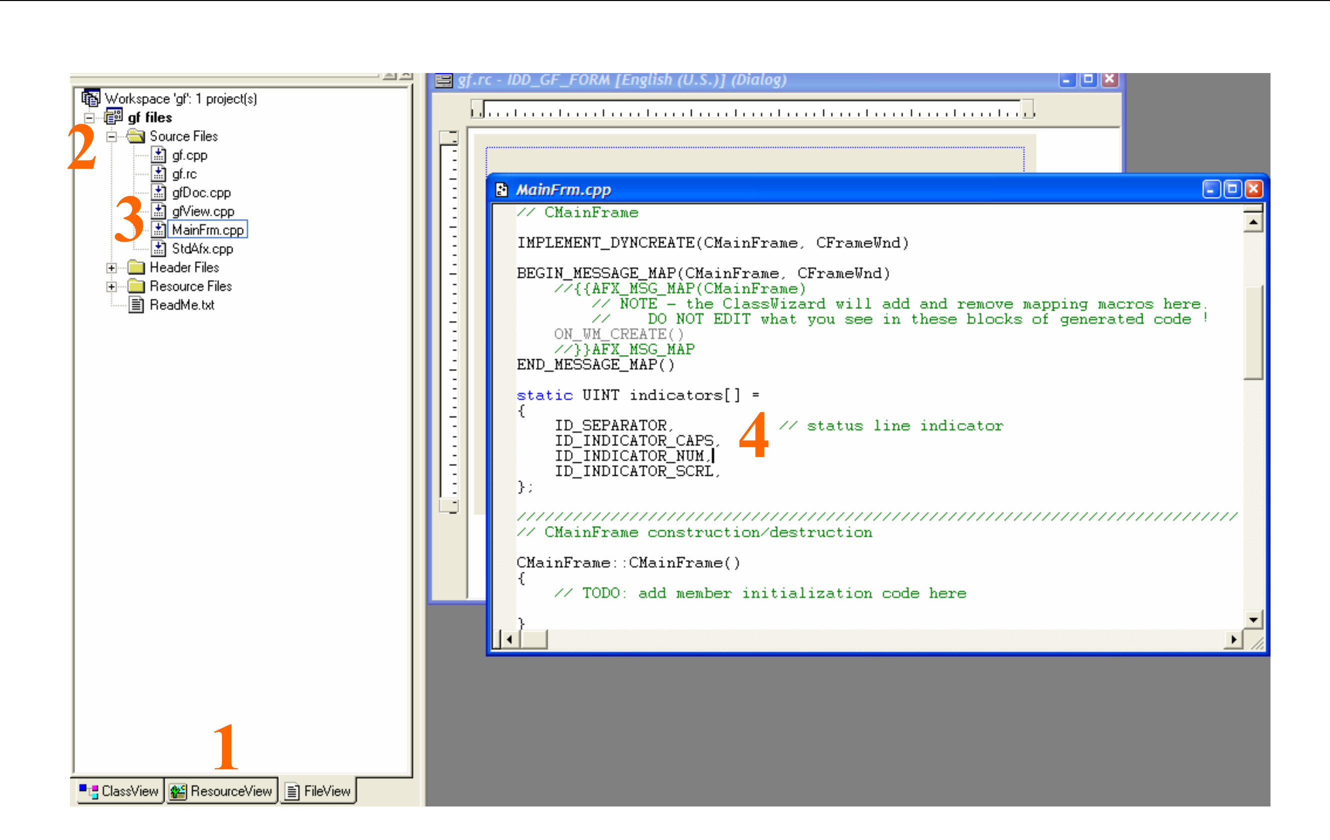
يوجد فيه ثلاث مفاتيح أساسية و هي

SCROLL, CAPS, NULL

* لاضافة مفتاح اساسي لشريط الحالة في قسم Indicator نقوم بالذهاب الى

ResourceView – SourceFile – MainFrame.cpp

يكون هنالك مصفوفة معرفة باسم UINT indicators[] و يمكن حذف مفتاح اساسي بتعليقه أو بشطبه . كما في الشكل التالي :


Workspace 'gf': 1 project(s)
gf files
Source Files
gf.cpp
gf.rc
gfDoc.cpp
gfView.cpp
MainFrm.cpp
StdAfx.cpp
Header Files
Resource Files
ReadMe.txt
ClassView
ResourceView
FileView
1
2
3
4
gf.rc - IDD_GF_FORM [English (U.S.)] (Dialog)
MainFrm.cpp
// CMainFrame
IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)
BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
//{{AFX_MSG_MAP(CMainFrame)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code !
ON_WM_CREATE()
//}}AFX_MSG_MAP
END_MESSAGE_MAP()
static UINT indicators[] =
{
ID_SEPARATOR, // status line indicator
ID_INDICATOR_CAPS,
ID_INDICATOR_NUM,
ID_INDICATOR_SCRL,
};
// CMainFrame construction/destruction
CMainFrame::CMainFrame()
{
// TODO: add member initialization code here
}

** **Files**

الملفات

* هنالك ثلاث طرق للتعامل مع الملفات

1. Database
2. C Archive
3. CFile

من خلال الطريقة الثالثة تستطيع أن تعمل ما يلي :

- Open File
- Read File
- Write File
- Create File
- Rename File
- Move File
- Delete File

* كيفية إنشاء ملف من CFile

```
CFile A;
A.Open ("اسم و مسار الملف", CFile:: modeCreate);
```

* كيفية الكتابة على ملف

```
CFile A;
A.Open("اسم و مسار الملف", CFile ::modeWrite);
A.Write ("النص",عدد الرموز );
```

يجب أن يكون الملف منشأ مسبقاً .

* كيفية إعادة تسمية ملف

```
CFile A;
A.Open ("اسم و مسار الملف ", CFile:: modeRename);
A.Rename ("اسم الملف الاول ","اسم الملف الجديد ");
A.Close ();
```

* كيفية القراءة من ملف

يجب تعريف متغيير من نوع char لتخزين الحروف في الملف
و متغير من نوع UINT لتخزين موقع الحرف
حيث أن UINT نوع بيانات غير محدد .

```
UINT x;
char c [100000];
CFile A;
A.Open (, CFile::modeRead);
x=A.Read(c, 100000);
c[x] =0;
MessageBox(c);
A.Close ();
```

* كيفية حذف ملف

```
CFile A;
A.Remove ("اسم و مسار الملف");
```

* **Printing**

الطباعة

Crystal Report:

- في حالة ربط المشروع مع قاعدة البيانات .
- في لغة ++VC توفر لنا بيئة الطباعة عند إنشاء المشروع .

* الدوال التي ينفذها برنامج ++VC من مرحلة صدور أمر الطباعة حتى نهاية العملية هي :

1. On_Begin_Printing () : يتم استدعاء هذه الدالة بعد ظهور مربع حوار الطباعة . في هذه الدالة يتم تحديد مواصفات النص المراد طباعته من حيث نوع الخط حجم الخط .... الخ .

2. On_End_Printing () : يتم استدعاء هذه الدالة بعد إغلاق مربع حوار الطباعة و إنهاء الطباعة للأيعاز للجهاز بأن عملية الطباعة قد انتهت ، ويتم تفريغ دالة On_Begin_Printing () .

3. On_PrepareDC () : دالة مسؤولة عن التأكد من أدوات الطابعة حيث تتأكد من أن الطابعة تقوم بالطباعة دون مشاكل .

4. On_PreparePrint () : هذه الدالة لربط القيم الابتدائية للطباعة .

5. On_Print () : دالة مسؤولة عن توجيه نصوص الى الطابعة وهي المسؤولة عن إستدعاء دالة (On_Draw ()) لطباعة محتويات صفحة الشاشة .

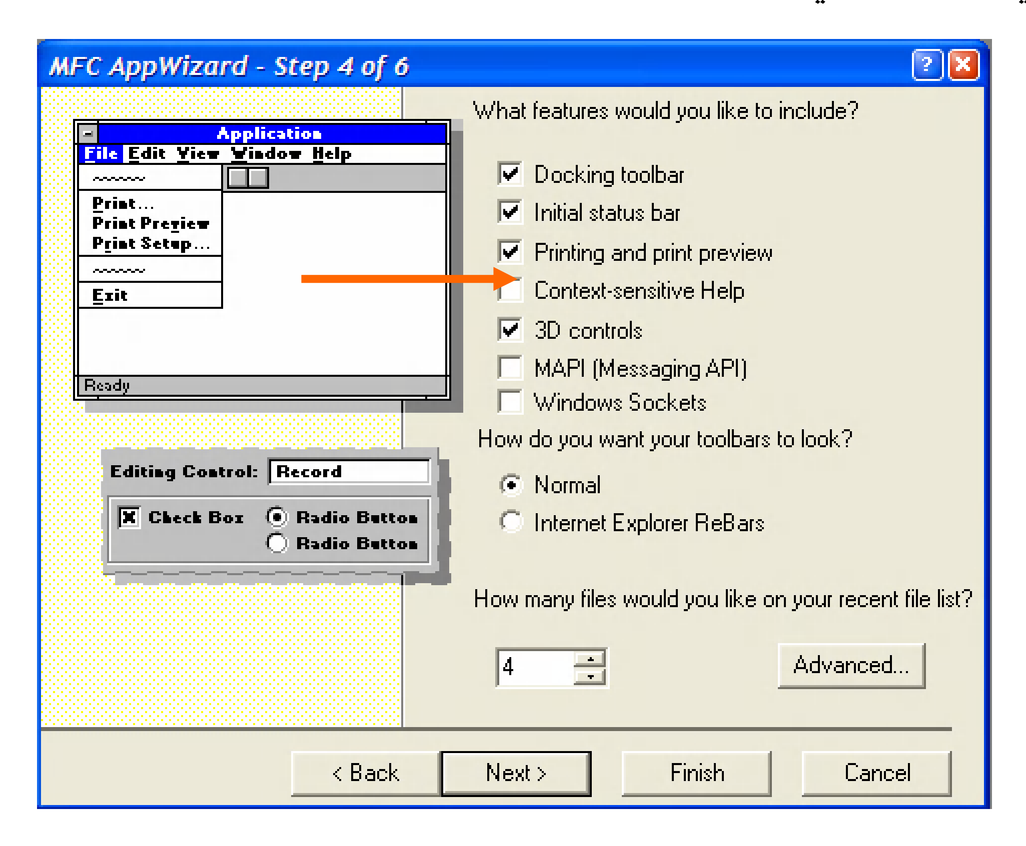
* **Help Files**

ملفات المساعدة

و هي ملفات مساعدة للمستخدم تعرض شرح توضيحي للأدوات المستخدمة في لغة VC++ ، نستطيع التعديل على ملفات المساعدة و حذف ملف مساعدة ، و إضافة ملف مساعدة .

يتم عمل ملفات المساعدة بطريقتين هما :

* الطريقة الأولى :
إنشاء ملف المساعدة مع المشروع (خلال مرحلة الانشاء) في الخطوة الرابعة من مرحلة الانشاء بحيث يتم إختيار Context-sensitive Help كما في الشكل التالي :



* الطريقة الثانية :

إلحاق (تصميم) ملف المساعدة بعد إنشاء المشروع .
يكون إمتداد ملف المساعدة (rtf.)

** أجزاء ملف التعليمات :

1. 1. العنوان : يكون بين << >> .
2. القوائم الرئيسية : وهي قوائم رئيسية يتفرع منها قوائم فرعية و تكون بشكل Link .
3. القوائم الفرعية : و هي الصفحة التي تظهر نتيجة إختيار أحد الخيارات في القائمة الرئيسية .
4. إختيار Context-sensitive Help .

** ملفات التعليمات

1. AFXcore.rtf : ملف تعليمات يحتوي على جميع المساعدات الاساسية الموجودة في البرنامج .
2. AFXPrint.rtf : ملف يحتوي على جميع التعليمات و المساعدة لعملية الطباعة .

** Hcw :
ملفات تقوم بتحويل ملفات التعليمات لملف تنفيذي ينفذ عند إستدعاء Help وهي ثلاث ملفات بإمتداد (h / .cnt / .w.) .

** كيفية إنشاء ملف التعليمات

إنشاء ملف تعليمات بعد مرحلة إنشاء المشروع كما يلي :

1. شريط العنوان يكون بالشكل التالي :

<< VC++ >>

2. القوائم الرئيسية :
   - اسم القائمة الرئيسية #
   - كتابة كل خيار في سطر منفصل
   - الضغط على مفتاح Ctrl + Enter عند نهاية كل سطر وهي فاصل الصفحات (جعل كل خيار Link لصفحة أخرى) .
3. القوائم الفرعية :
   - اسم عنصر القائمة الرئيسية
   - إضافة الشرح اللازم للصفحة .
4. حفظ الملف بإمتداد rtf.

** Active X control

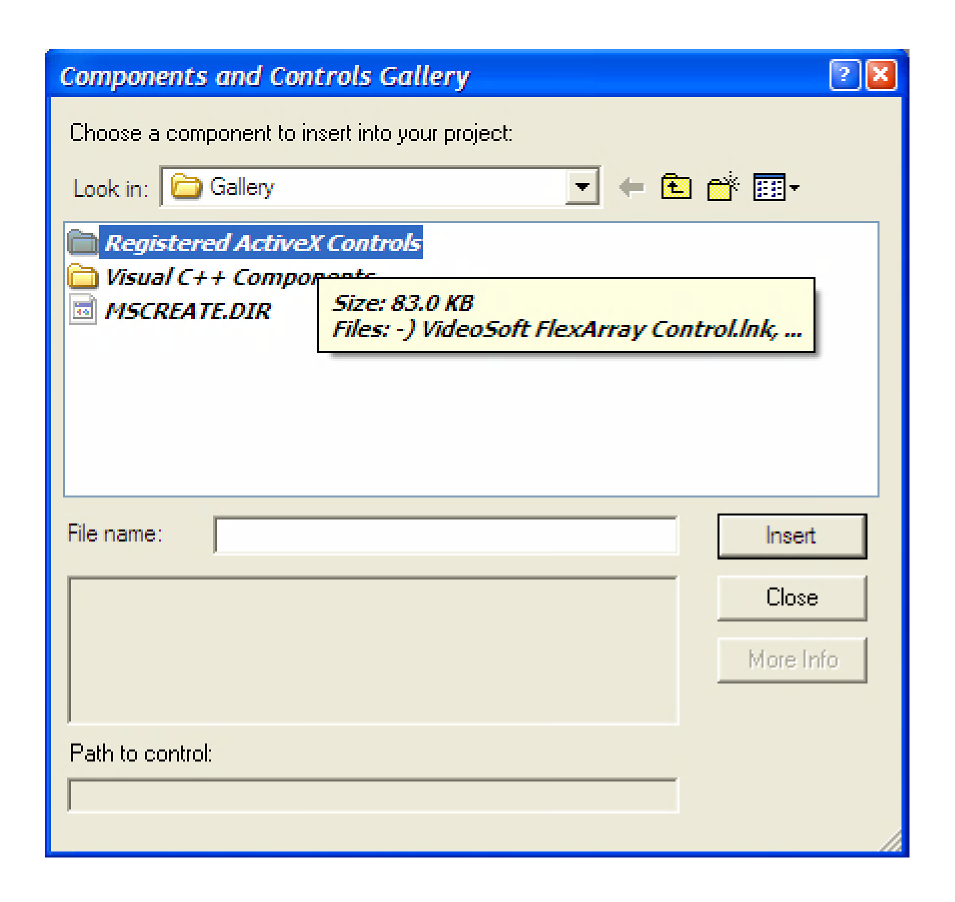
عبارة عن أدوات مبنية جاهزة في بيئة ++VC و تحتوي على مكتبات خاص بها (حيث تحتوي هذه المكتبات على جميع الدوال الخاصة بالاداة)
حيث أن أدوات ++VC الأساسية تعمل على نظام 16 bit
و أدوات Active X Control تعمل على بيئة نظام 32 bit أو 64 Bit
و أمتداد ملف Active X يكون (.OCX)

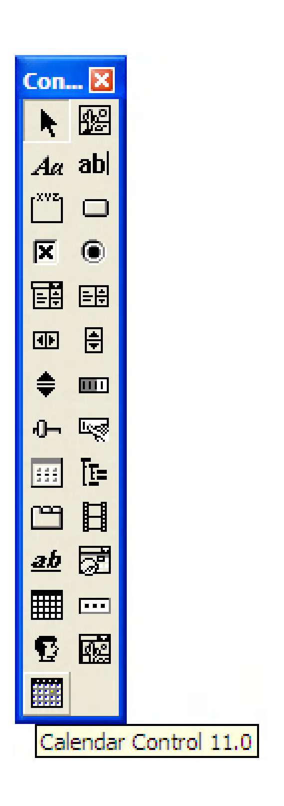
* الادوات المطلوبة من Active X هي

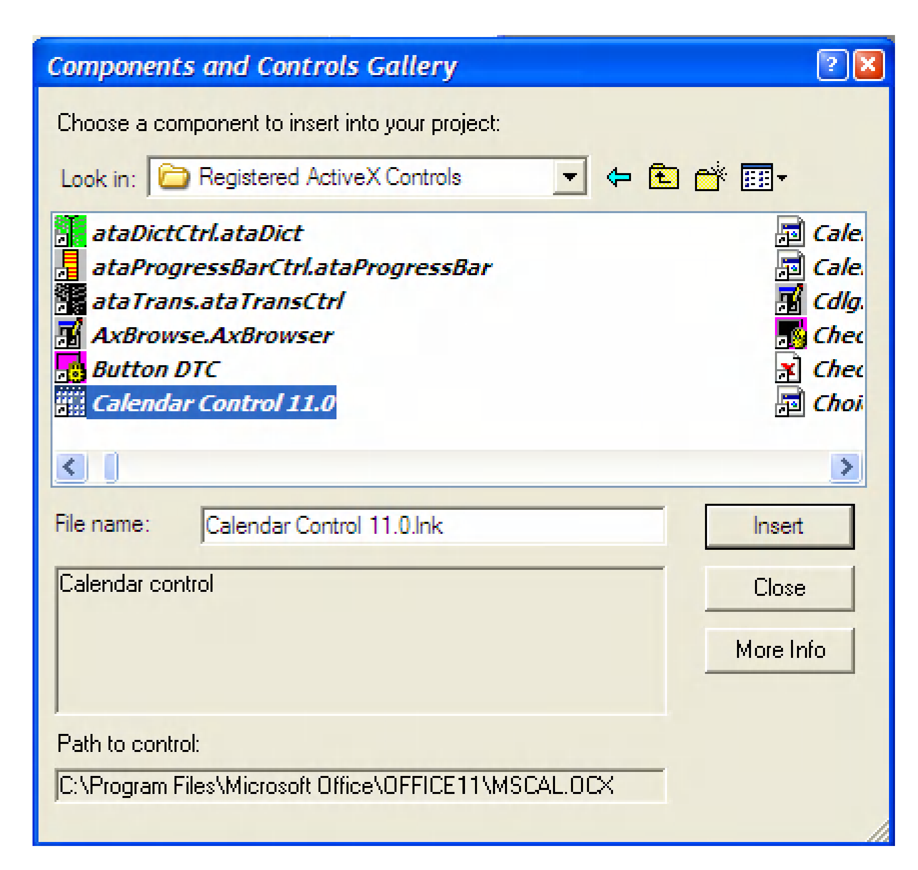
1. Grid الشبكة
2. Calendar التقويم

* خطوات إضافة ملف Active X الى المشروع

1. يتم نقل الاداة الى مجلد system 32 في محرك الاقراص :C .
2. يتم الذهاب الى Run من قائمة Start و كتابة الامر التالي
C:\WINDOWS\system32\regsvr32.exe calendar.ocx
3. في برنامج ++VC نختار قائمة Project ومنها نختار Add To Project و منها أمر Component And Control فيظهر صندوق حوار كما في الشكل التالي :
من مجلد الـ Gallery نختار Registered ActiveX Controls و منه نختار Calendar Control 11.0 ثم نضغط على زر أمر Insert
تابع الصور التالية :


Components and Controls Gallery
Choose a component to insert into your project:
Look in: Gallery
Registered ActiveX Controls
Visual C++ Components
MSCREATE.DIR
Size: 83.0 KB
Files: -) VideoSoft FlexArray Control.lnk, ...
File name:
Insert
Close
More Info
Path to control:


Con...
Calendar Control 11.0


Components and Controls Gallery
Choose a component to insert into your project:
Look in: Registered ActiveX Controls
ataDictCtrl.ataDict
ataProgressBarCtrl.ataProgressBar
ataTrans.ataTransCtrl
AxBrowse.AxBrowser
Button DTC
Calendar Control 11.0
File name: Calendar Control 11.0.lnk
Calendar control
Insert
Close
More Info
Path to control:
C:\Program Files\Microsoft Office\OFFICE11\MSCAL.OCX

مثال : أنشئ شبكة Grid تتكون من 6 أعمدة و 8 صفوف

- الصف الاول و العمود الاول عبارة عن عناوين
- العمود الاول أرقام من 0-7
- الصف الاول أرحرف من A-F

يتم تعبئة الخلية عند تغيير موقع المؤشر :

نقوم بالذهاب الى :

View – ClassWizard – IDC_GRID1 – SelChange – Add Function – EditCode.

* يجب تعريف متغير من Grid

```
void CDsgView::OnSelChangeDbgrid1(short FAR* Cancel)
{
   CString str;
   G.SetCol (0);
   for (int i=0; i<8;i++)
   {
       G.SetRow (i);
       str.Format("%d",i);
       G.SetText (i);
   }
   G.SetRow(0);
   for (int j=1;j<6;j++)
   {
       G.SetCol(j);
       str='A'+(j-1);
       G.SetText(str);
   }
}
```

أخيراً ....

أهدي كتابي هذا الى التي بها نُظمت هذه الخاطرة

عينيها

في عينيّها الشمسُ و القمرِ
كونٌ بديعُ الخلقِ
عالمٌ لم يمسسهُ إنسنٌ من قبلْ
عالمٌ لطالما حلمتُ فيه
كأن الشمسُ مُنحدرةٍ
مستقرةٌ في عينها اليسرى
إحتار عقلي في إختيار البقاءِ
بإحدى عينيّها
لأن الأخُرى مستقرٌ فيها القمرِ
إحتار عقلي فحكمتُ قلبي
قال لي :
رويداً و لما الحيرةُ أيُها العاشقِ
فلتبقى على خطِ الأستواءِ بين عينيّها
و لتكون في اليسرى الشمسُ
و في الأخُرى القمرِ
فالشمسُ شمسُ الشتاءِ دافئةٌ في يومكَ المُثلجِ
و القمرُ قمرٌ مضيئٌ في ليلك الأسودِ
و لما الحيرةُ أيُها العاشقِ
فلتبقى على خطِ الأستواءِ بين عينيّها أبدَ الدهرِ
أيُها العاشقِ ......

<A>

في النهاية

هذا كل ما أعلم و الله عز و جل أعلى و أعلم ، و أقف عند هذه النقطة حتى أبدء من عندها بموضوع أخر بأذن الله تعالى .

أرجو أن أكون قد وفقني الله تعالى في سرد مواد هذا الكتاب المتواضع و أود من الله تعالى أن ينفعكم و لو بالقليل في أعمالكم .

اللهم أني أعوذ بك من علم لا ينفع و من نفس لا تشبع ومن قلب لا يخشع ومن دعوة لا يستجاب لها أمين .

كل الشكر لمن علمني عالم البرمجة و أخص بالذكر مثلي الأعلى الأستاذ : **سنان أحمد نصار**

راجياً من الله تعالى فيكم الدعاء لي بالتوفيق بظاهر الغيب .



مع أطيب الأمنيات

أخوكم

عبدالكريم محمد أبونمر

Jordan / Irbid

Phone: 00962788507656

E-Mail: abunemer1987@yahoo.com